

AD-A086 444

DOTY ASSOCIATES INC ROCKVILLE MD

F/6 9/2

HANDBOOK OF PROCEDURES FOR ESTIMATING COMPUTER SYSTEM SIZING AN--ETC(U)

FEB 80 W B HUMPHREY, J N POSTAK

F19628-79-C-0106

UNCLASSIFIED

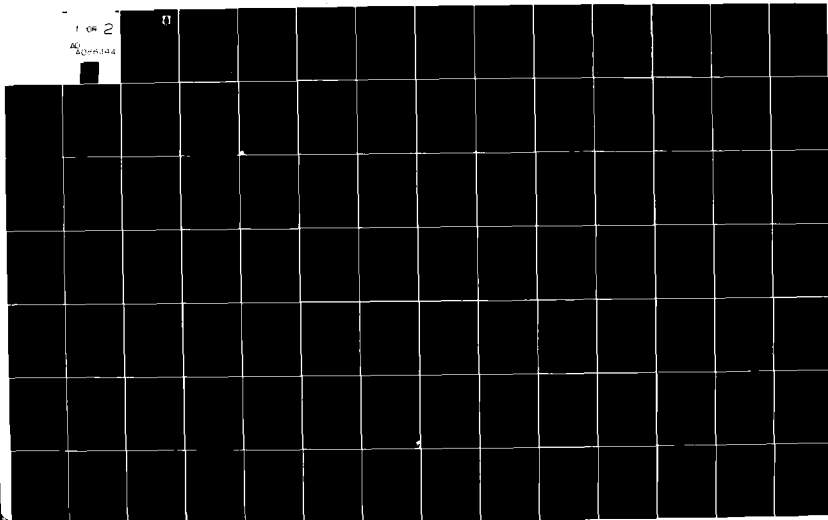
DAI-TR-235-VOL-1

ESD-TR-80-115-VOL-1

ML

1 OF 2

AD-A086 444

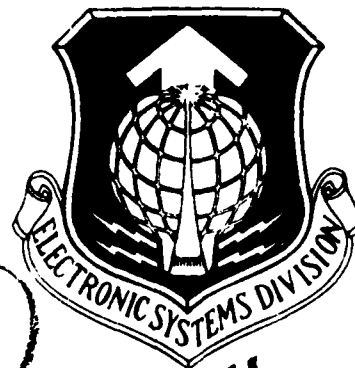


HANDBOOK OF PROCEDURES FOR ESTIMATING
COMPUTER SYSTEM SIZING AND TIMING
PARAMETERS

Volume I: Procedures and Techniques

LEVEL III

12



Doty Associates, Inc.
416 Hungerford Drive
Rockville, Maryland 20850

15 February 1980



Final Report for Period 16 May 1979 - 15 February 1980

Approved for Public Release;
Distribution Unlimited.

Prepared for

DEPUTY FOR TECHNICAL OPERATIONS
ELECTRONIC SYSTEMS DIVISION
HANSCom AIR FORCE BASE, MA 01731

UDC R-5 COPY

80 7 1 083

LEGAL NOTICE

When U. S. Government drawings, specifications or other data are used for any purpose other than a definitely related government procurement operation, the government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

OTHER NOTICES

Do not return this copy. Retain or destroy.

REVIEW AND APPROVAL

This technical report has been reviewed and is approved for publication.

Neil D. McQuage

NEIL D. McQUAGE, Major, USAF
Project Manager

James W. Neely, Jr.

JAMES W. NEELY, Jr., Lt Col, USAF
Acting Chief, Technology Applications
Division
Deputy for Technical Operations

FOR THE COMMANDER

C. J. Grewe, Jr.

CHARLES J. GREWE, Jr., Lt Col, USAF
Acting Director, Computer Systems
Engineering
Deputy for Technical Operations

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ESD/TF-PP-111 - Vol-1	2. GOVT ACCESSION NO. AD-A086444	3. RECIPIENT'S CATALOG NUMBER 2
4. TITLE (and Subtitle) HANDBOOK OF PROCEDURES FOR ESTIMATING COMPUTER SYSTEM SIZING AND TIMING PARAMETERS. Volume I. Procedures and Techniques.		5. TYPE OF REPORT & PERIOD COVERED Final Technical Report 10 May 1974 - 15 May 1974
6. AUTHOR(s) William H. Humphrey John N. Postak		7. PERFORMING ORG. REPORT NUMBER DAI-TP-111-1
8. CONTRACT OR GRANT NUMBER(s) F14-74-1-116		9. PROGRAM ELEMENT PROJECT TASK AREA & WORK UNIT NUMBERS 474 F 12/132
10. PERFORMING ORGANIZATION NAME AND ADDRESS Ray Associates, Inc. (DAI) 400 Somerset Drive (311) 404-070 Buckville, Maryland 2105		11. CONTROLLING OFFICE NAME AND ADDRESS Electronic Systems Division (ESD/TOI) Air Force Systems Command (AFSC) Hanscom Air Force Base, Massachusetts 01731
12. REPORT DATE 15 May 1974		13. NUMBER OF PAGES 163
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) DAI-TP-111-1		15. SECURITY CLASS (of this report) Unclassified
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		17. SECURITY CLASS (of this report) Unclassified
18. SUPPLEMENTARY NOTES Major Neil D. McQuage, USAF Electronic Systems Division (ESD/TOI) Air Force Systems Command (AFSC) Hanscom Air Force Base, Mass. 01731 (617) 861-2925		19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Acquisition Life-Cycle Bibliography Command, Control and Communications (C3) Systems Computer System Sizing Computer System Timing (continued next page)
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This first volume of the handbook presents procedures that provide a structured approach for estimating computer system sizing and timing parameters during the acquisition life-cycle of Embedded Computer Systems (ECS). The procedures were designed for use by the Electronic Systems Division's Computer Systems Engineering Directorate (ESD/TOI). Based on engineering discipline, the procedures provide a step-by-step program to assist ESD/TOI engineers and computer scientists in evaluating or (continued next page)		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

406573

MP

19. Key Words (continued)

Computer Systems
Data Item Description
Embedded Computer Systems
Estimations
Glossary
Hardware
Hardware Sizing
Management Procedures
Management Techniques
Metrics
Quality Assurance
Software
Software Engineering
Software Sizing
Software/Hardware Tradeoffs
System Configuration
System Engineering
System Environment
System Performance

20. Abstract (continued)

conducting initial sizing and timing estimates, updates, or developmental monitoring. Procedural steps are discussed with emphasis on actions that should be taken, risks that should be considered, constraints that may be encountered, and factors that may affect the quality of the estimate at various stages of the acquisition. Techniques that can be used in estimating sizing and timing parameters are also discussed, including data requirements, assumptions, and levels of confidence.

In addition, this volume also contains a reference listing, bibliography, glossary, and an index.

The second volume of the handbook is an addendum that contains supplemental illustrative information regarding selected ESD C3 Embedded Computer Systems (ECS), sets of graphic representations of sizing and timing relationships of computer systems, two examples of the application of the procedures presented in this volume, a proposed revision of the Data Item Description (DID) entitled, Computer Program Timing and Sizing Data (DI-S-30568), a list of suggestions for future work to update and improve the estimating procedures, a reference listing, an annotated bibliography, and an index.

PREFACE

This handbook of procedures for estimating computer system sizing and timing parameters during the acquisition life-cycle of Embedded Computer Systems (ECS) was developed for use by the Electronic Systems Division's Computer Systems Engineering Directorate (ESD/TOI). Based on engineering discipline, the procedures provide a step-by-step program to assist ESD/TOI engineers and computer scientists in evaluating or conducting initial sizing and timing estimates, updates, or developmental monitoring. Procedural steps are discussed with emphasis on actions that should be taken, risks that should be considered, constraints that may be encountered, and factors that may affect the quality of the estimate at various stages of the acquisition. An introduction to techniques that can be used in estimating sizing and timing parameters is presented, including data requirements, assumptions, and levels of confidence.

The handbook consists of two volumes; this first volume discusses procedures and techniques, and the second volume provides an addendum of supplemental information. The primary features and tools of this handbook are as follows:

Volume I

Procedures
Techniques
Factors and Checklists
Glossary
References
Bibliography
Index

Volume II

Case Studies
Hardware Specifications
Graphs
Data Item Description
Abstracts
Index

Development of this handbook was accomplished under Contract No. F19628-79-C-0106 by Doty Associates, Inc. (DAI). Inclusive dates of the technical effort were 16 May 1979 through 15 February 1980.

Dwy Associates, Inc. (DAI) would like to acknowledge the excellent assistance and support provided by the Air Force Technical Contract Monitor, Major Neil D. McQuage, USAF of ESD/TOI and his staff: Captain Mary K. Welty, 2ND LT. Ralph P. Dunlevy, and 2ND LT. Henry A. Hughes. Their responsiveness to inquiries, comments on draft documentation reviews, and overall cooperation during this technical effort were greatly appreciated.

Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DDC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/ _____	
Availability _____	
Dist	Special
A	

CONTENTS

Section		Page
1	INTRODUCTION.	1
1.1	Background	1
1.2	Goal and Content	3
1.3	Organization of this Volume	4
2	ACQUISITION LIFE CYCLE	5
2.1	Introduction	5
2.2	A Baseline System Acquisition Process	6
2.3	ECS Software and ECS Acquisition Fluidity	8
3	ESTABLISHMENT OF THE ECS REQUIREMENTS	11
3.1	Developing and Monitoring ECS Requirements	14
3.2	Requirements Considerations and Problems Areas	15
3.2.1	Conceptual Phase	16
3.2.2	Validation Phase	17
3.2.3	Full-Scale Development Phase	17
4	OVERVIEW OF DEVELOPMENT OF ECS SIZING AND TIMING ESTIMATES	19
4.1	Sizing and Timing Approach	20
4.2	Study Framework	27
4.2.1	Purpose of the Sizing and Timing Study	27
4.2.2	Initial Resource Estimates	27
4.2.3	Comparison of Alternative Concepts	28
4.2.4	Preparation of Procurement Documents	28
4.2.5	Proposal Evaluation	29
4.2.6	Development Monitoring	29
4.2.7	Impact of Changes in Requirements or Workload	30
4.2.8	System Performance Evaluation	31
4.2.9	Conclusion	31
4.3	Sizing and Timing Techniques	32
5	CONDUCTING SIZING AND TIMING STUDIES.	35
5.1	Study Overview	35
5.2	Conducting Initial Sizing and Timing Studies	38
5.2.1	Set Objectives	39
5.2.2	Define System Boundaries and Workload	39
5.2.3	Establish Candidate Techniques	41
5.2.4	Identify and Qualify Data Sources	42
5.2.5	Prepare a Detailed Study Approach	43
5.2.6	Conduct the Sizing and Timing Study	44
5.2.7	Correlate Findings With Other Program Data	44
5.2.8	Analyze Study Results	45
5.2.9	Prepare the Study Report	45
5.2.10	Corrective Action	46

CONTENTS (continued)

<u>Section</u>	<u>Page</u>
5.3 Study Update	47
5.3.1 Review Prior Study	48
5.3.2 Verify Study Objectives	48
5.3.3 Adjust System Boundaries and Workload	49
5.3.4 Verify and/or Modify Technique(s) Selection	49
5.3.5 Verify and/or Modify Data Sources	50
5.3.6 Update Detailed-Study Approach	50
5.3.7 Conduct Study Update	50
5.4 Developmental Monitoring	51
5.4.1 Prepare Prediction Set	53
5.4.2 Verify Analysis Techniques	55
5.4.3 Receive Data or Request Update	55
5.4.4 Update Preliminary Data File	55
5.4.5 Perform Analysis	56
5.4.6 Prepare Report	56
5.4.7 Update of Data File	57
5.4.8 Take Corrective Action	57
6 INTEGRATION OF SIZING AND TIMING	59
6.1 Sizing and Timing as a Discipline	59
6.2 Spare Memory Capacity	60
6.3 Sizing and Timing Data Collection	61
 TAB A INTRODUCTION TO SIZING AND TIMING TECHNIQUES	 A-1
 REFERENCES	 REF-1
BIBLIOGRAPHY	BIBLIO-1
GLOSSARY	GLOSS-1
ACRONYMS AND ABBREVIATIONS	ACRO-1
INDEX	INDEX-1

LIST OF FIGURES AND TABLES

<u>Figure</u>	<u>Title</u>	<u>Page</u>
1	Baseline System Acquisition Process	7
2	Overview of Sizing and Timing Study Logic Flow	22
3	Requirements and Workload Relationship	23
4	Conceptual C3 Embedded Computer System (ECS)	25
5	Throughput Vs. Delay Curve	26
6	Sizing and Timing Techniques	33
7	Sizing and Timing Study Procedures	36-37
8	Sizing and Timing Initial Study Procedures	38
9	Sizing and Timing Update Procedures	47
10	Sizing and Timing Developmental Monitoring Procedures	51
11	Example of a Completed Prediction Set Form	54
A-1	Sizing and Timing Techniques	A-2

<u>Table</u>	<u>Title</u>	<u>Page</u>
1	Factors to be Considered Prior to Acceptance of a System Design.	12-13
2	Topics for Study Approach	44
3	Topics for Study Report	46

1. INTRODUCTION

1.1 Background.

For each new generation of Command, Control, and Communications (C3) systems, Embedded Computer Systems (ECS) become increasingly critical. There are many reasons for this; computer systems, including software, are performing more functions more rapidly, and they must be reliable under more extreme physical and operational constraints. As more stringent requirements have been imposed upon Command, Control, and Communications (C3) system performance (e.g., more targets and target types, more sophisticated signal processing, improved countermeasures and counter-countermeasures, faster responses, and greater reliability), increasing demands have been placed upon the C3 computer systems.

In many system acquisitions, the uncertainties of qualifying and quantifying the hardware and software needs of an ECS have been extensive, with software uncertainties far exceeding those of the hardware. This situation is projected to increase in frequency and severity in the future. The requirement to perform most necessary functions in real-time is very demanding and has thus provided the impetus for continued improvement in hardware and software technologies. However, in spite of these technological improvements, increased operational requirements continue to dictate larger and more sophisticated computer systems and software packages. As a consequence, there is an increasing, if not urgent, need to develop and evaluate embedded computer systems sizing and timing estimates.

The field of computer system acquisition management and engineering is just developing, especially in its ability to derive sizing and timing estimates for entire computer systems and software in the overall computer system environment. Sizing and timing of whole or large segments of computer systems are particularly complex and, at the present state-of-the-art, only appear feasible using models, simulations, benchmarks or monitors. The primary difficulty is the definition of a workload that a

computer system or subsystem must perform. Additionally, the following difficulties in sizing and timing computer systems have been acknowledged:

- lack of standard engineering and management procedures and techniques,
- lack of standard metrics used in developing estimates,
- lack of understanding of the relationships between hardware and software characteristics and operational requirements,
- lack of accurate or timely projections of hardware and software resource requirements, and
- lack of awareness of viable system design and technological advances.

The Department of Defense, recognizing these deficiencies, established the DoD Management Steering Committee for embedded computer resources. As stated in the background section of the charter for the group in DoD Directive 5000.29:

...Current annual expenditures by the Department of Defense on the design, development, acquisition, management and operation support of computer resources embedded within and integral to weapons, communications, command and control, and intelligence systems are measured in the billions of dollars. At the same time, such computer resources have often presented critical cost and schedule problems during the development and acquisition of new defense systems. Even after system implementation and fielding, the software has often proven unreliable...

Computer system resource estimating has historically been characterized by two shortcomings; it has been poorly done and seldom validated. The reasons for poor estimating are numerous. The lack of necessary information resources to implement reliable estimating methodologies is among the most important; also:

- There is no common data base from which to develop computer resource estimates.

- Even where some historical computer systems data exist, there is often no clear understanding of what the data actually represent.

The problems associated with embedded computer system sizing and timing estimates are more pronounced and diverse than those associated with stand-alone systems. When a C³ system is developed with integrated embedded computers, there is often concurrent hardware/software development. An immediate problem, therefore, is that there may be no functioning hardware on which to begin software integration. The concepts of early defect removal using modern programming practices may be less than adequate to prevent severe software or hardware problems when the system is completed. Estimates can be done by use of models, simulations, benchmarks or monitoring; however, these techniques may not be feasible due to costs, time constraints, or lack of relevant data.

The above discussion of the background highlights some of the major problems and concerns regarding estimation of computer system sizing and timing parameters. It also demonstrates the critical need for Air Force managers to acquire standardized engineering and management procedures, now presented in this handbook, to insure that the operational requirements of C³ ECS are met during the system acquisition life-cycle in the most efficient and effective manner.

1.2 Goal and Content.

The main emphasis of this handbook is to provide standard procedures rather than metrics for estimating sizing and timing parameters of ECS in Air Force C³ systems, for use by the engineers and computer scientists of the Electronic Systems Division's Computer Systems Engineering Directorate (ESD/TOI). Such use should enable the engineers and computer scientists of ESD/TOI to better assist other personnel at ESD in conducting sizing and timing studies. IN THE EVENT ESD PERSONNEL NEED ASSISTANCE IN CONDUCTING A SIZING AND TIMING STUDY - CONTACT ESD/TOI.

This handbook consists of two volumes. This first volume contains the procedures and techniques, and the second volume provides an addendum of supplemental information. In this first volume, the procedures structure the function of estimating ECS sizing and timing parameters into three types: initial studies, study updates, and developmental monitoring, and provides a step-by-step program employing many of the techniques used in Verification and Validation (V&V) of systems. The procedural steps are discussed with emphasis on actions that should be taken, risks that should be considered, constraints that may be encountered, and factors that affect the quality of an estimate at various stages of the acquisition. Techniques that can be used in estimating sizing and timing parameters at various stages of the acquisition are discussed, and include general information regarding required data, assumptions, application, and level of confidence. In the second volume, information about selected ESD C3 ECS is presented to illustrate the types of data that could be considered in the development of computer system analogies for sizing and timing studies. Also included are a number of graphically illustrated computer system sizing and timing relative relationships that are based on algorithms developed by various computer scientists. In addition, there are two examples that discuss the application of the procedures.

1.3 Organization of this Volume.

The organization of the remainder of this volume is as follows: Section 2 discusses the acquisition life-cycle of ECS; Section 3 discusses the establishment of ECS requirements; Section 4 provides an overview of the development of ECS sizing and timing estimates; Section 5 presents the step-by-step procedures for conducting sizing and timing studies; Section 6 discusses the integration of sizing and timing; Tab A provides information on sizing and timing techniques; and in addition, there is a reference listing, bibliography, glossary, and an index for use with this handbook.

2. ACQUISITION LIFE-CYCLE

2.1 Introduction.

In order to address the problems and solutions associated with the estimation of ECS sizing and timing parameters, it is helpful to review the acquisition life-cycle. The process of system acquisition has evolved over a relatively short time. The late 1960's and early 1970's provided the first real efforts towards bringing system discipline to the acquisition process with the establishment of the Defense Systems Acquisition Review Council (DSARC), the Blue Ribbon Defense Panel, and the promulgation of Department of Defense Instruction (DODI) 5000.1.

One of the problems that has plagued the acquisition of embedded computer systems is that the management technology and software engineering skills have failed to keep pace with, and in fact have never been equal to, the complexity of embedded computer systems. These inadequacies have been discussed by numerous authors. The following list of references are but a few: 2/, 3/, 5/, 8/, 9/, 12/, 14/, 15/, 17/. This dilemma is further compounded by the proliferation of management "guidance," and in the ever-increasing number of studies aimed at improving the acquisition process. There is in fact more guidance than can be successfully applied to any acquisition. There exists a problem of determining what techniques and controls should be applied as opposed to what must be applied.

The current baseline for the acquisition of major systems is the DSARC process. 4/ In 1976 in response to recommendations made by the Commission on Government Procurement, the Office of Management and Budget (OMB) issued Circular A-109, Major Systems Acquisition. In response to this circular, DoD Instruction 5000.1 and DoD Instruction 5000.2 are being revised to clarify DoD policy. The major shift in the emphasis of policy concerning major system acquisitions is in the front end of the acquisition cycle. Two specific areas are validation of the mission need and the increased use of competition. It is recognized that not all cases of

acquisition or modification of embedded computer systems will be designated as major systems, but, as will be discussed in Section 5, it is critical that additional emphasis be placed on the front-end work of any ECS if true improvement in the quality of sizing and timing estimates is to take place.

2.2 A Baseline System Acquisition Process.

Figure 1 shows the acquisition process for both hardware and software, which are components of embedded computer systems. This process is an idealized version and in practice is rarely followed exactly as displayed. Paragraph 2.3 discusses some of the many conditions that cause a deviation from this process, and the associated impact on sizing and timing estimates. This is not to say that there should not be deviations from the acquisition life-cycle, in fact AFR 800-14 Volume II states that a program may skip phases, or may have concurrent activities in any or all phases.

The Conceptual Phase in a major acquisition begins with the determination by the Secretary of Defense that a mission need is essential. This determination is the Milestone 0 decision, program initiation. The primary thrust of the Conceptual Phase is along three lines. First, there is the administrative process of establishing a program manager with sufficient authority to execute the second major conceptual effort: that of investigating alternative design concepts and recommending the Validation Phase structure, or even whether or not to proceed with Validation at all. The third thrust of the Conceptual Phase is the development of the acquisition strategy. The development of the strategy provides the foundation for the acquisition plan and assists the program manager in defining, as is presently known, the path that a program will take. The ultimate direction of a program should depend upon the results obtained in the Conceptual and Validations Phases.

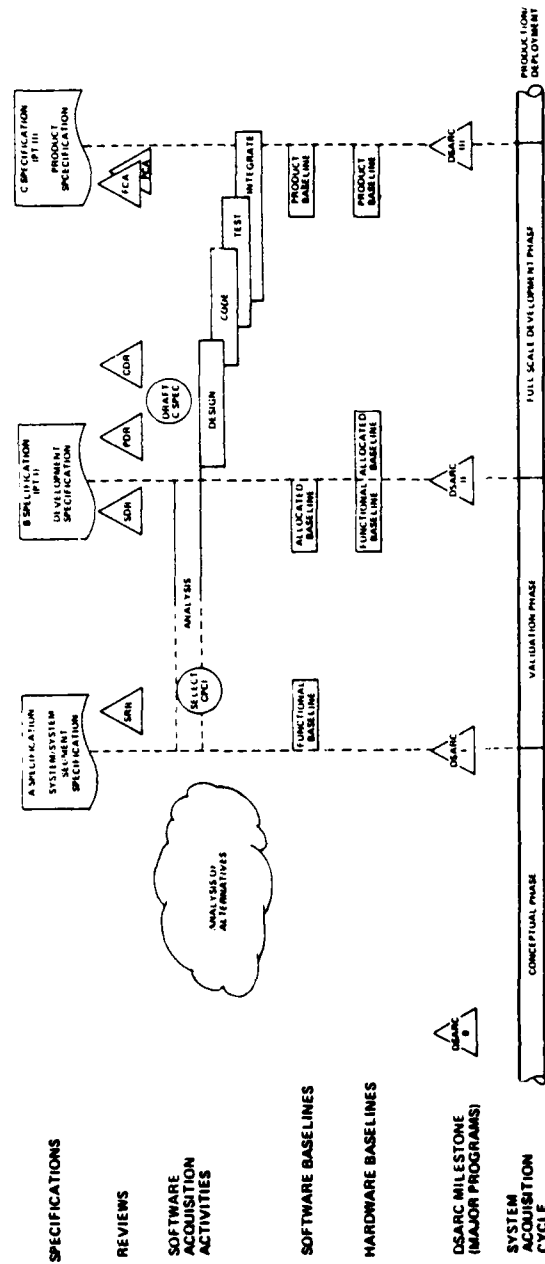


Figure 1. Baseline System Acquisition Process

During this phase all sources available for the development of competing concepts should be considered; industry, non-profit organizations, government laboratories and educational institutions. Parallel short-term study contracts, coupled with experimental validation of high risk or innovative concepts should be the rule. With the increased emphasis on competition and alternative approaches, the primary product of the Conceptual Phase, the system/system segment specification (Type A), becomes even more critical.

Following the Conceptual Phase in a major acquisition is the Validation Phase. This phase is designed to provide reduction in risk to the point where the next phase may be commenced, to select the best alternative system, and to continue to develop the acquisition strategy by soliciting and evaluating proposals. This phase, for both major and minor acquisitions, is highly critical and as is discussed later, is often either disregarded or combined with the Conceptual Phase, with Full-Scale Development following the combined Conceptual/Validation Phase. The product of the Validation Phase is primarily the B Specification (Part I Development). During this phase the System Requirements Review (SRR) and System Design Review (SDR) are conducted, and the Allocated Software Baseline and Functional Hardware Baselines are established.

The final phase in the acquisition process is the Full-Scale Development Phase, where the critical software construction efforts occur, the Allocated Hardware Baseline is established, followed by the Hardware and Software Product Baselines. The Full-Scale Development Phase is started by the DSARC II decision and involves a major commitment of funds for the complete engineering development of the system, the procurement of long-lead items, and the many tasks associated with preparing for production.

2.3 ECS Software and ECS Acquisition Fluidity.

The process for the development of a system containing an embedded computer system is by its very nature one that is full of tradeoffs,

redirections, and competing goals, and is subject to a host of external pressures. The actual acquisition process for a major system is rarely as outlined in OMB Circular A-109. In some cases, an upgrade to an existing system may result in more technical problems than those encountered in the acquisition of a new system. There are a few basic combinations of the acquisition or modification of a system with embedded computer resources, but there will always exist the possibility of a new twist, another combination, that will result in a deviation from the acquisition plan, no matter how well constructed. For example a new threat may require the backfitting of a capability that was never considered, or one that has consciously been excluded, which is even more difficult. Some of the possible conditions that may exist for a new system acquisition are:

- Concurrent hardware and software development is required.
- Adequate hardware is available and only software development is required.
- Adequate software is available and only hardware development is required.
- Only an upgrade of existing hardware and software is required.

Once the baseline structure is established, there exists the problem of phasing the activities within the process. The acquisition process taken from a macro approach should be orderly and manageable, but rarely is. Therefore, the procedures and techniques developed herein are based on two major premises:

1. That certain factors must be known at certain points in the acquisition process to support reasonable sizing and timing estimates.
2. That in the event these factors are not known, steps may be taken to:
 - a) estimate the missing factors,
 - b) provide a measure of risk in not having the actual data, and
 - c) provide for the accelerated acquisition of the missing data.

It is not acceptable to assume that just because a System/System Segment Specification is delivered, that the delivery constitutes an appropriate increase in the system definition. The development of sizing and timing estimates must be viewed as a continual refinement of initial estimates, progressing towards a completely defined, developed, and operating system, to the maximum extent possible in the system's acquisition life-cycle.

3. ESTABLISHMENT OF THE ECS REQUIREMENTS

The establishment of the ECS requirements is the single-most important key to the successful development of a C³ system. 10/

This statement may be considered obvious by some, and absolutely wrong by others, but the fact that there is disagreement on the major cause of problems associated with the development of C³ systems should be a matter of concern. There is a definite gap in the ability of system developers to adequately translate total system requirements into ECS requirements. Often the information required to adequately size a system will not be known until the system is developed. 10/

In addressing the ECS requirements of a C³ system, there are many objectives that must be evaluated. Based on Martin 13/, a listing of factors that should be considered prior to accepting any system design is presented in Table 1. It is unlikely that all the items will be fully answered until the system is actually complete, but the items are primarily those that must be considered, or at least with which project personnel must be most familiar. If specific data can not be obtained, then there must be a reliable method to alert the key program office personnel of the potential impact of not having the data, and also a method that will schedule the time when the data will become available. In a competitive environment there will be different detailed approaches to solve the system problem, but to be able to indicate the level of detail available from competing offerors, a common format will be of assistance.

Table 1 is by no means exhaustive, nor is it possible to have all the data prior to committing to a system design. What is important to note is that sizing and timing estimates are a part of the system design and cannot be developed with any degree of accuracy outside the knowledge base, such as the factors listed in Table 1, that bounds workload, equipment, and total project resources. The more accurate the data

TABLE 1. FACTORS TO BE CONSIDERED PRIOR TO ACCEPTANCE OF A SYSTEM DESIGN

IMPACT ON ESTIMATION OF		FACTORS
SIZING	TIMING	
		<u>REQUIREMENTS DEFINITION</u>
H	H	Current system configuration (if applicable)
H	H	Degree of real-time operation
M	H	Functional capabilities
L	L	Number of operators
M	M	Reliability/Maintainability
L	L	System locations
M	H	System throughput
M	H	System turnaround time
		<u>SYSTEM WORKFLOW</u>
M	H	Anticipated average CPU utilization
M	M	Anticipated changes in workload
M	M	Anticipated size of outputs
L	M	Average workload
M	M	Background jobs
M	H	Critical workload
H	M	Definition of permissible degradation
M	H	Degree of multiprogramming
H	M	Hardware and software utilization, by job
M	H	Job arrival distributions
M	H	Job arrival rates
L	L	Job class definition
L	M	Job priorities
M	H	Job service times
L	M	Maximum I/O rates anticipated
H	M	Number of output reports
L	L	Override capabilities
M	H	Peak workload
L	M	Permissible off-line jobs
M	H	Probability of meeting critical period workloads
H	H	Required reserve processing capacity
M	M	Scenario definition
L	L	Security requirements, by job
L	H	System timing diagrams
L	M	Testable scenarios and jobs
		<u>HARDWARE REQUIREMENTS</u>
M	M	Buffers
H	H	Candidate computer configurations
H	H	Characteristics of CPU
N	H	Communication line speeds
M	M	Hardware functions in firmware
N	L	Hardware monitors

LEGEND	
H	High Impact
M	Medium Impact
L	Low Impact
N	Negligible Impact

TABLE 1. FACTORS TO BE CONSIDERED PRIOR TO ACCEPTANCE OF A SYSTEM DESIGN
(continued)

IMPACT ON ESTIMATION OF		FACTORS
SIZING	TIMING	
		<u>HARDWARE REQUIREMENTS</u> (continued)
M	M	I/O handlers
L	H	I/O speeds
L	M	Input device configurations
H	M	Memory configuration
H	M	Memory expansion capability
L	M	Output device configurations
		<u>SOFTWARE REQUIREMENTS</u>
H	M	Application programs
H	H	Compilers
M	M	Complexity
N	N	Development facilities available
N	L	Error Correction
L	H	Module execution times
H	H	Operating system architecture
N	L	Program flow chart
L	L	Programming languages
L	M	Reliability
H	L	Size estimate of developed software
M	M	Software monitors
M	M	Utilities
		<u>FILE STRUCTURE</u>
M	L	Accounting files
L	M	Addressing methods
N	N	Data security classification
L	L	File protection
M	M	File structures
H	L	Historical data requirements
M	M	Off-line storage
H	H	On-line storage
H	L	Predicted file growth
M	M	Type of data storage
M	M	Temporary job files
M	L	Test file data

LEGEND
H=High Impact
M=Medium Impact
L=Low Impact
N=Negligible Impact

obtained to the above factors, the better will be the estimate of system sizing and timing parameters. Table 1 may be used as an initial checklist to assess the knowledge available for input to the estimation process.

3.1 Developing and Monitoring ECS Requirements.

The development of ECS requirements is an extremely difficult process that often becomes even more difficult as the acquisition life-cycle progresses. Several factors contribute to this increasing difficulty:

- Early requirements are viewed as "estimates only" which will be assumed to change anyway.
- The ECS requirement is not the system being developed, but only a portion of the electronic C3 system that is being developed to meet a requirement.
- The further into the acquisition life-cycle a program progresses, the shorter the lead time available to modify earlier decisions, which may have been incorrect.
- There still exists a feeling that in the event of an incorrect hardware decision, problems may be corrected in software.
- Requirements do change, even well into Full-Scale Development, and it becomes harder to reestablish the desired balance among the elements of the ECS and available resources.

The degree of uncertainty of the ECS requirements should ideally decrease with increased baseline definitions. However, in many cases, and all too often, initial estimates of the ECS requirements fail to consider that there does exist a degree of uncertainty in what will be the final configuration. It has been shown by Herd ^{9/} that initial estimates on program size may be in error by 200% in the conceptual stage. Early estimates tend to be viewed as more accurate than they are. Later in the process when all the indicators suggest that there may be a problem with the software size, people tend to deny the existence of the problem.

One of the reasons that ECS sizing and timing problems are recognized too late is that there is no uniform application of a methodology to

establish, budget, and track compliance with the initial and revised estimates. There are several Data Item Descriptions (DIDs) that provide for the submission of such estimates, but these are not routinely applied throughout ESD acquisitions. The critical factor is not so much that a particular DID is used or not used, but that the information is not obtained. Further analysis of specific DIDs is contained in Section 6. In addition to the submission of estimates by contractors, Program Office personnel and Technical Representatives should conduct periodic on-site reviews of the contractor's backup data for their sizing and timing estimates. These on-site reviews should be coordinated with regularly scheduled program reviews that should occur at least every three months.

3.2 Requirements Considerations and Problem Areas.

The requirements problems associated with embedded computer systems within C³ systems are complicated by several factors:

- C³ systems are inherently complex because of the random nature of workloads and requirements.
- C³ systems are usually "one of a kind," and therefore there exists no large data base from which to extrapolate data to be used in estimates for future systems.
- C³ functions for a given system are subject to large changes because of the changing nature of the threats with which the military must deal.
- There exists no catalog of approved applications software designed for a function. In addition, there are no standards for defining functions. This is analogous to designing new integrated circuits for each new application.
- Recent trends in the procurement of major weapon systems dictate that many technical approaches should be evaluated in response to a government requirement. The process permits direct comparisons of cost and schedules of proposed approaches, but makes the comparison of technical system parameters difficult. If competing contractors were bidding to detailed specifications, unrealistic parameters of ECS sizing and timing would be easier to identify, in that the extremes of a data group would be apparent.

- Generally it is the large major contractors who are involved in C3 system acquisition, and this in itself further reduces the number of approaches that may be compared.
- The length of time to acquire major systems further complicates the problem of refining estimates, because long periods of time pass between estimates and demonstrated achievements.
- Since much of the hardware for C3 systems may be in existence, frequently the time allotted for Request For Proposal (RFP) preparation is too short.

These are some of the factors that result in less than adequate requirements definition. Failure to adequately define requirements will haunt a program to the bitter end.

3.2.1 Conceptual Phase. The Conceptual Phase in the system acquisition life-cycle is the time during which alternative concepts are developed and evaluated to determine which will be responsive to established mission requirements. With respect to computer resources, AFR 800-14 Volume II, Acquisition and Support Procedures for Computer Resources in Systems, states:

...(the Conceptual Phase) is the initial planning period where the technical military, and economic bases are established through comprehensive studies, experimental development and concept evaluation. ...the major definitive document resulting from this phase is the initial system specification which documents total system performance requirements.

It is acknowledged that the Conceptual Phase is a highly iterative process, and therein lies one of the major problems associated with sizing and timing of ECS systems in the early stages. The problem is that estimates of software size are often used as the input for estimating initial resources required in justification for transition into the Validation Phase, and yet, software size is also an output which must form the basis for the operational system in meeting the requirement. There are numerous competing objectives that must be treated separately as well

as in concert. Some critical factors to recognize are the purpose of the sizing and timing estimate, the degree of accuracy possible, and above all the assumptions that go into the estimate, as discussed in Section 5.

The output of the Conceptual Phase, the System Specification, developed in accordance with MIL-STD 483 (as modified by ESD guidance), is critical to successful program completion. The degree of detail in the system specification should not preclude the possibility of alternate solutions in subsequent phases, and yet to provide even reasonable estimates of ECS development costs for the Program Decision (entry into the Validation Phase) there must be detailed definition of the software, even to the level of definition of file size and structure, data output, and the relationships among other programs. The alternative is to use macro estimating techniques, recognizing that large uncertainties are involved, and be prepared to accommodate the worst-case situation.

3.2.2 Validation Phase. In keeping with the increased emphasis on retaining competition in the early acquisition phases and the exploration of several alternative approaches, the Validation Phase has taken on increased importance. Yet historically, this phase has often been bypassed. Whether or not the Validation Phase is omitted and Full-Scale Development follows the Conceptual Phase, the fact remains, the degree of definition provided by validation will still have to be done! There is no way around the requirement to progress from an ill-defined system to a working system except by increasing one's knowledge of the system. Two critical reviews that should be conducted during the Validation Phase are the System Requirements Review (SRR) and the System Design Review (SDR). MIL-STD-1521A provides guidance on the conduct and requirements of the SRR and SDR.

3.2.3 Full-Scale Development Phase. The Full Scale Development Phase (FSD), including limited production, is initiated when the need of the system has been reaffirmed, and the work of the preceding stages indicates the soundness of the selected approach. The software Allocated Baseline

is to be complete at this point, and the hardware Allocated Baseline shortly after the start of FSD. The primary purpose of the FSD is the design, fabrication and testing of the system under development. The output of this phase is a system, closely resembling the production system, that has been tested and documented sufficiently to permit a production go ahead.

The FSD phase contains the most labor-intensive effort of the acquisition: actually designing, coding, testing and integrating the software. During this phase is the specifications transition from the development to the product specification. The major reviews in this phase are the Preliminary Design Review and the Critical Design Review.

In a large, one-of-a-kind system, there may not be a "production" of a system, but only a militarization of parts of the system. From the viewpoint of sizing and timing, two very critical activities take shape. First, monitoring the ECS development and second, conducting the test and evaluation activities. When the test plans and procedures are being developed it is critical to retain the trail from requirements to workload, to test conditions, and finally to test procedures.

4. OVERVIEW OF DEVELOPMENT OF ECS SIZING AND TIMING ESTIMATES

There exists no agreed-upon definition of sizing and timing estimates. The generation of estimates of parameters of ECS sizing or timing is a process, with the result being a deterministic figure that is in appropriate units corresponding to the parameter that was chosen to be measured. System sizing parameters could be any or all of the following, or more:

- number of lines of source or object code required for the operating system software,
- number of lines of source or object code required for the application software,
- number of software functional modules determined to be included in the above,
- number of lines of object code required to be resident in the main memory at any one given time,
- amount of data and programs that can be maintained offline,
- number and types of Input/Output (I/O) terminals that are required for proper operation of the system,
- capacity required for the main memory in order for the system to perform within the time constraints of the operating environment, and
- capacity of the mass memory that is required for storage of all necessary data and programs.

This points out that when discussing sizing of an ECS system, one must determine what segment of the ECS is specifically being addressed in order to even begin to discuss the subject.

System timing is an equally complex process to bound. Timing may have different meanings, depending upon the circumstances and conditions of the discussion. Examples of timing parameters are:

- elapsed time occurring between the end of a query input at a terminal and the start of the response output,
- time required for the execution of a software module's processing action,
- throughput time required to complete the necessary processing of an estimated peak workload,
- queuing times estimated for each I/O device, memory or CPU,
- time required to update various data bases,
- utilization time of CPU or other components that is determined for a specific workload, and
- time required to transfer data through communication lines or networks.

Once again the specific time condition to be estimated must be defined. In addition to defining the particular sizing or timing estimate, the purpose of the estimate may dictate what technique or combination of techniques might be most applicable (see Section 4.3).

4.1 Sizing and Timing Approach.

The primary parameter of any sizing and timing study is the purpose for which the study is being conducted. This initial entry point in the process was chosen for the following four reasons:

- The functional purpose of the study is not dictated by the phase of the acquisition life-cycle, rather by the procedures and data available.
- The acquisition life-cycle of C³ systems with embedded computer systems will normally vary with each system, therefore attempting to dictate a particular study approach in a particular acquisition life-cycle phase would severely limit utility.
- The quality of the system definition and the quality of documentation are subject to wide variations.

- Since the procedures must be applicable to all ESD acquisitions, they must permit accommodation of a wide range of system definitions, for example, in some cases hardware may be dictated, in others not.

Coupled with the purpose of the study and candidate procedures, is the overall logic behind any sizing and timing study. Here again, the requirement that the procedures be applicable to all acquisition life-cycle phases, and to all ESD C³ acquisitions, dictates that a sound, adaptable study logic be developed, based on engineering disciplines. The logic diagram for the sizing and timing studies proposed herein is shown in Figure 2. This logic is similar to the work conducted by Gilbert, et al., of the Federal Computer Performance Evaluation and Simulation Center (FEDSIM). ^{7/} Additional emphasis has been placed in two areas: first, in an expansion of the effort to permit utility in the early phases of the acquisition life-cycle; and second, an increased emphasis on measurement of the uncertainty associated with the results of the study.

The concept of integrating the sizing and timing estimates within a single study framework is based upon the common denominator of WORKLOAD. The workload of the ECS drives the system configuration and results in a given system performance. The workload is the transition variable between system requirements and system performance, and yet is extremely difficult to quantify early in an acquisition. Figure 3 illustrates the relationships between requirements, workload, the ECS, and the C³ system. The ECS must respond to the workload imposed by the C³ system requirements. If the workload does not support the requirements, the ECS can not support the C³ system and will result in inadequate outputs or poor system performance.

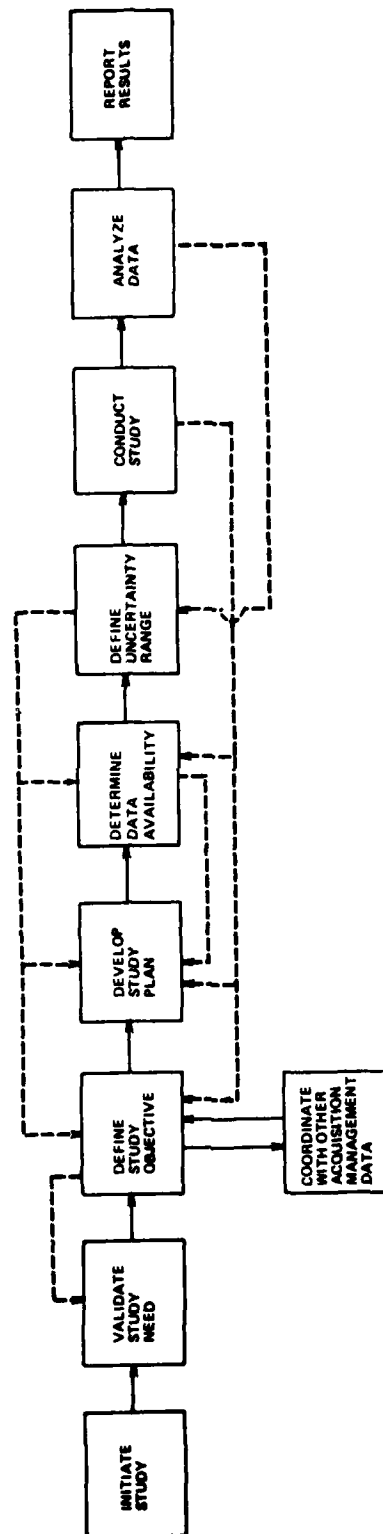


Figure 2. Overview of Sizing and Timing Study Logic Flow

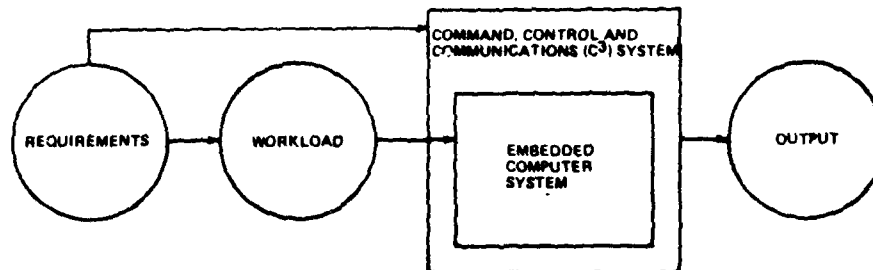


Figure 3. Requirements and Workload Relationship

The workload of an embedded computer system as used in this report is defined as the total demand placed upon the embedded computer system by the user in a specified period of time.

Much has been written about the lack of requirements definition being a major contributor to the development of systems that exceed cost targets and fail to meet performance requirements. This is undoubtedly true; however, another large contributor has been the lack of a workload definition as the bridge between requirements and system design. For example, a detailed breakdown of application software, on a module-by-module basis, may provide an accurate measure of ECS storage requirements, but would contribute little by itself to the problem of system timing. In addition, even the execution time of each compiled module would be of no benefit unless the sequence of execution for specific workloads was provided. Accordingly, one of the purposes of this Handbook is to integrate the conduct of sizing and timing studies with the concept of workload definition.

AFR 800-14 Volume II addresses several components of workloads, such as "minimum iteration rates for various functional processing," (para 3-4 f(4)), "functions are arranged in their logical sequence so that any

specified operational usage of the system can be traced in an end-to-end or in a closed-loop path," (para 4-5a(1)). In addition, paragraphs 4-9c and d require analysis of critical timing requirements and run-times at the Preliminary Design Review (PDR), and review of processing time and memory estimates at the Critical Design Review (CDR). Both of these requirements have workload implications. Also paragraph 5-6a discusses computer program Verification and Validation, wherein it states that during the analysis phase, "a timing and sizing study should be conducted to insure that the proposed computer system is adequate."

In order to complete the overview of ECS sizing and timing it is necessary to provide guidelines as to what the output of the studies will be, in well defined units. It would not be possible to provide a list of all sizing and timing output measures. The primary reason for this is that the output is determined or dictated by the objective of the study. For example, the primary measure of size could be words of main memory, or number of terminals, or even the character capacity of a Cathode Ray Tube (CRT). Therefore, as used in this Handbook, size is defined as:

The size of an embedded computer system has two components;
1) physical size: in terms of the number and types of physical hardware units and the number of unique software entities including data and their corresponding number of source statements, object words, or required storage space as appropriate; and 2) capacity: in terms of storage or processing capability of hardware that is not dependent upon software.

The precise boundaries of an embedded computer system will vary depending on the specific system. A conceptual C³ ECS configuration is provided in Figure 4. The number of components and interfaces will vary from system to system.

The timing of an ECS is also subject to wide variation and is dependent upon whatever the individual conducting the study has established as the objective. The following definition of timing of embedded computer systems forms the basis for the remainder of this Handbook:

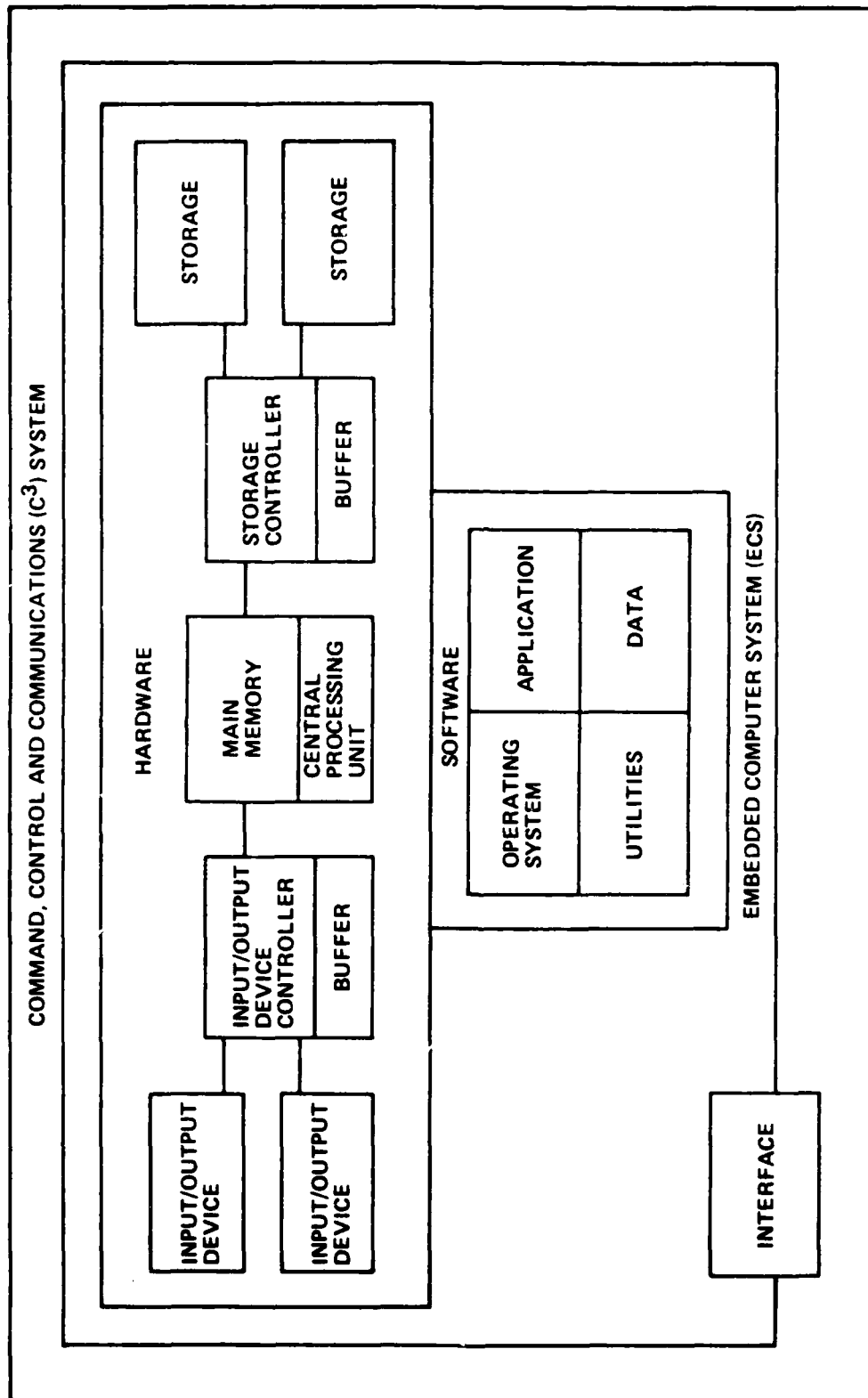


Figure 4. Conceptual C³ Embedded Computer System (ECS)

The timing of an embedded computer system has two components; 1) workload independent parameters such as memory access time, cycle time and printer output rate, and 2) workload dependent measures such as *THROUGHPUT*, the work completed per unit time for a given workload, *RESPONSE TIME*, the elapsed time between the finish of a request and the start of the output for a given workload, and *UTILIZATION*, the ratio of the time spent by a device performing work during a specified interval to the total interval, at a given workload.

There are many other derived parameters that may be considered measures of system performance. However, the concept of throughput, response time, and utilization answers three basic questions regarding an entire ECS or a specified component:

- How much work is being done?
- How fast is it being done?
- How much more work can the system do?

Beizer 1/ has stated that every system has a characteristic curve that relates response time (delay) to throughput similar to that shown in Figure 5.

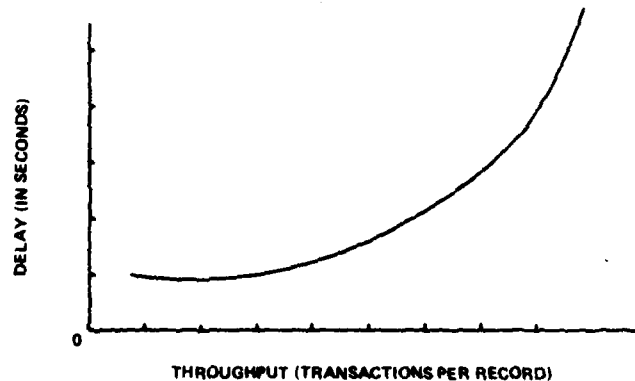


Figure 5. Throughput Vs. Delay Curve

It must be recognized that throughput and response time (or delay) are extremely complex for both a given system and a given workload, and are composed of such factors as job mix, job-arrival rate, job-processing time, data base, code construction, and more. The specific components of throughput and delay are discussed further in Section 5.

4.2 Study Framework.

4.2.1 Purpose of the Sizing and Timing Study. Determining the purpose of any particular sizing and timing study is the initial entry point for the start of the sizing and timing process. Seven potential points, during the acquisition life-cycle of ECS, at which sizing and timing studies would be beneficial, are discussed in sections 4.2.2 through 4.2.8. It is possible, and in some cases desirable, that more than one type of study be conducted at the same time. In this event, the multiple studies should be defined in a single study objective as discussed in Section 5.2.1 of this Handbook.

4.2.2 Initial Resource Estimates. The Conceptual Phase of the acquisition life-cycle is normally when initial estimates will be required. However, cases do arise when a new system may be required to interface with an existing system that is currently in Full-Scale Development, and the sizing and timing study may have to consider the existing constraints of the more mature system, but this is not the norm. A primary purpose for conducting an early sizing and timing study is to provide a basis for derived resource requirements needed to support the budget process. Nearly all models and techniques for estimating resources for the development of software use a measure of the estimated size of the software as the independent variable when computing manpower, schedule, computer usage, and hardware size. A key feature of any estimates derived from initial studies is that they should be used to measure limiting conditions of the overall proposed acquisition. For example, one could question whether the estimated code could reasonably be developed within the overall program schedule and manning. An estimate of system sizing and

timing is required when considering the initial system configuration. The recent emphasis on encouraging alternate system concepts, unconstrained by hardware, only increases the need for a more formalized approach to conducting sizing and timing based on general hardware configurations. The sizing and timing study conducted in response to configuration questions can provide limits on basic system parameters such as maximum processing speeds available from a given central processor, or required processing speeds for given program size and workloads. All program resources must be considered in relation to each other with the goal of reducing the possibility of having unattainable goals for hardware, software, time, and dollars.

4.2.3 Comparison of Alternative Concepts. The use of sizing and timing studies in support of the evaluation of alternate concepts presents an additional set of problems. In this instance, where the workload definition may differ even in response to the same requirement, the problem of comparing like performance indicators is compounded. The major difficulty is weighing the utility of various performance parameters, specifically throughput and response time. For example, one might consider the marginal utility of increasing throughput and decreasing response time. The other factor that enters the equation of system comparison is cost. Although not a specific output of a sizing and timing study, the estimates of size, as described above, are normally used as the basis of cost estimates. There are additional factors that are independent of workload, such as memory access speed, that will enter the comparison.

4.2.4 Preparation of Procurement Documents. Sizing and timing studies conducted in support of the preparation of Requests for Proposals (RFP) can be invaluable, in that conducting such studies will assist in the process of translating system requirements into workloads. This in turn will provide a better basis for structuring Statements of Work (SOW) and the development of the evaluation of proposed systems. There exist competing objectives for specifying enough mandatory requirements, while

not biasing the RFP in favor of a particular system. In some cases, however, if the acquisition is a modification to an existing system, the preexisting constraints may severely limit the options of responding contractors.

4.2.5 Proposal Evaluation. The evaluation of competitive and sole-source proposals for complex systems is an extremely difficult and time-consuming process. The Source Selection Evaluation Board (SSEB) must be knowledgeable in areas of technical documentation, the associated cost proposals, and also must be concerned with the evaluation criteria and contractual aspects of the proposal evaluation. With the increasing emphasis on alternative solutions, the direct comparison of different proposals can be difficult at best. By conducting a sizing and timing study from proposals, based on the common definitions of sizing and timing parameters as defined in this Handbook, the competing capabilities may be evaluated. This is based on the requirement, however, that in the RFP guidance for proposal preparation, the basic components of the sizing and timing input data were requested. An advantage of conducting a formal sizing and timing study during proposal evaluation is that relative comparisons of throughput and response time may be made from the study workload as opposed to discrete numbers which may or may not have been derived from identical workload assumptions.

4.2.6 Development Monitoring. The use of sizing and timing studies during development of an ECS can support other management reporting systems such as Cost/Schedule Control Systems Criteria (C/SCSC) reporting. C/SCSC reports, as provided by the Cost Performance Report (CPR) for major systems, are primarily concerned with the financial status of the program and with the concept of earned value; that is, what costs were budgeted for completed work. The obvious difficulty is that it is very hard to convert parameters such as system sizing and timing into elements of cost. One possible solution is to plan for and budget for the contractor to conduct increasingly refined sizing and timing studies and use the quantifiable reduction in sizing and timing estimate errors as the

measurement criteria. All too often initial sizing and timing estimates are not refined until the start of integration. The fact that the execution time of the modules are known is no guarantee that the higher-level component will perform to satisfy the system requirement. Sizing and timing studies conducted by or reviewed by the Program Office (PO) must be done on a routine basis. The longer a system goes without a revision or update of a sizing and timing estimate, the more suspect the estimates become and the more potentially complex future corrections become, both in terms of the cost of corrections and decreased reliability of the system. The key is scheduled refinement of estimates with measurable verification.

4.2.7 Changes in Requirements or Workload. One of the few things that may safely be said about the acquisition of C3 systems is that there will be requirement changes. These changes may take three forms: modifications, deletions, and additions. Modifications to requirements may occur as a result of several factors, such as a clearer definition of what the requirements really were, as a result of new mission needs, or as a result of possibly competing objectives. The deletion of requirements is most likely to occur as a result of three factors: the cost to implement the requirements, the need to remain within performance constraints, and the need to meet an established operational target date. The addition of requirements is apt to take place early in the acquisition life-cycle as a result of user desired system enhancements. The system user tends to require more and more capability. The danger here is that the cumulative effect of small, early changes will not be translated into additional workload. Consequently, when the system is tested, its overall performance may surprisingly be found unacceptable.

Sizing and timing studies should be conducted to evaluate the impact of any change. The degree of complexity of the study is a function of the detailed knowledge of the system and the possible, not probable, consequence of the change. Of primary concern in conducting this type of study is the definition of the workload change. To properly evaluate require-

ments and hence workload changes, it is essential that the baseline workload be tracked throughout the entire acquisition life-cycle. Section 5 discusses a workload tracking and update system.

4.2.8 System Performance Evaluation. The measurement of system operation is conducted on the operational system, or portions thereof. It may be used to exercise the system against a benchmark program. The benchmark program may or may not describe the projected system workload; however, the benchmark workload has a specific meaning to the designer who may be interested in comparing specific features of two systems.

Discrete event simulators are also used to evaluate a system that may be experiencing unacceptable response time, where the cause of the bottleneck is unclear. The development of large-scale discrete simulators can be very costly, but may reveal system choke points, which may then be redesigned within the simulation and reverified for performance improvement.

4.2.9 Conclusion. It is important to understand the relationships between sizing and timing estimates, and the acquisition life-cycle. Initially system timing is a requirement, and the process of making timing estimates is for purposes of (1) insuring that the timing requirements (throughput, response time) are still what is required, and (2) that the system development progresses to meet those requirements. On the other hand, system sizing is an estimate of what will be required to meet the functional and timing requirements. A serious budgetary problem may occur in the acquisition of a C³ system. The initial ballpark sizing estimates provided for a budget may be assumed to be specific figures rather than estimates by financial management personnel. Accordingly, use of estimate figures should be identified and explained in budget submissions.

The relationships reverse as a system proceeds through development, in that the size becomes fixed as software is developed and hardware is

assembled. Actual timing must be estimated or measured to verify that the requirement is met. All too often in Full-Scale Development the timing requirements are relaxed because the size, and therefore budgets, cannot be expanded. The way to attack the problem is to continually monitor the relationships among sizing, timing, and resource parameters.

4.3 Sizing and Timing Techniques.

Sizing and timing techniques are presented in Figure 6. As indicated in the figure, techniques are divided into three classes: EXTRAPOLATION techniques that may be used in the early phases of acquisition when the sizing and timing parameters of the proposed ECS are difficult to define; REPRESENTATION techniques that may be used when more details are known about the sizing and timing parameters; and MEASUREMENT techniques that may be used in the latter part of the acquisition to determine actual sizing and timing parameters. Considering categories of techniques, there is no clear distinction as techniques move from ANALYTICAL to SIMULATION. Many authors discuss analytical models as discrete event simulation models. The real difference is the assumptions made in the model construction, particularly in the assumption of the distribution assigned to the workload. When dealing with a C³ ECS, an exact mathematical analysis of the ultimate actual workload may be impossible, and even assumptions of job arrival times may pose a real problem. Techniques are discussed further in Sections 5.2.3, 5.3.4, and 5.4.2, and Tab A.

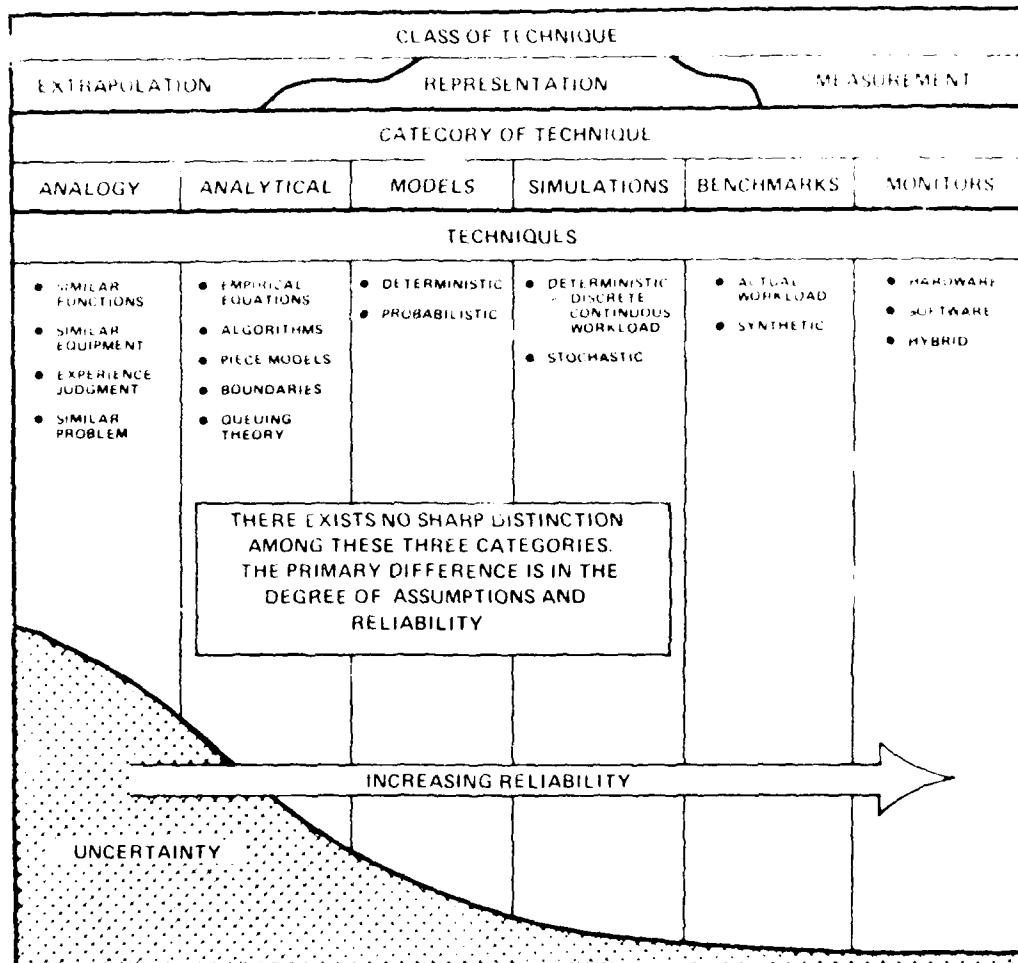


Figure 6. Sizing and Timing Techniques

5. CONDUCTING SIZING AND TIMING STUDIES

5.1 Study Overview.

The basis for improved estimates of ECS sizing and timing parameters lies in the application of engineering disciplines. By establishing and using standard procedures on a regular basis, estimators (users) will increase their understanding of sizing and timing studies. This understanding should improve the users' ability to conduct sizing and timing studies and to objectively analyze the studies conducted by others.

In addition to using these procedures on a regular basis when conducting sizing and timing studies, one should insure that all studies are adequately documented and retained for future reference in Program Offices and/or ESD/TOI.

The sizing and timing study procedures presented in this handbook are for use by the engineers and computer scientists of the Electronic Systems Division's Computer Systems Engineering Directorate (ESD/TOI). They, in turn, should be able to better assist other ESD personnel in conducting or evaluating sizing and timing studies. IN THE EVENT ESD PERSONNEL NEED ASSISTANCE IN CONDUCTING A SIZING AND TIMING STUDY - CONTACT ESD/TOI.

THE APPLICATION OF THE PROCEDURES DEPENDS ON HOW MUCH IS KNOWN ABOUT A SYSTEM RATHER THAN THE SYSTEM'S ACQUISITION PHASE. Availability of sizing and timing data will vary depending on such factors as: the extent of in-depth data contained in the system's definition and specifications (for example see Table 1), the sizing and timing data collection requirements imposed upon the contractor, and the aggressiveness of the Program Office.

Efforts or studies to estimate the sizing and timing parameters are divided into three different types (see Figure 7):

- Initial Studies are usually conducted in the early stages of the conceptual system definition activity.

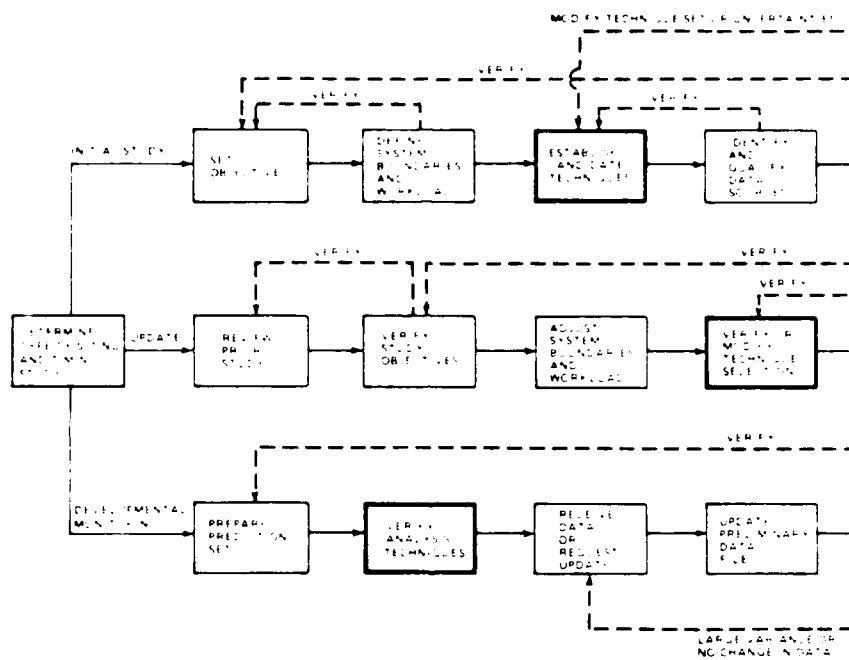


Figure 7. Sizing and Timing Study Procedures Logic Flow

- Update Studies should be conducted at critical points throughout the system acquisition life-cycle. These studies should be done when any significant changes occur that might impact on sizing and timing parameters (also see Table 1 in Section 2), prior to major program reviews (also see Figure 1 in Section 2), or any other time an update is deemed appropriate. This latter event might well be on the occasion of key personnel changes so that new personnel will have an understanding of how prior studies were conducted and accordingly develop their own level of confidence in the developed estimates.
- Developmental Monitoring consists of studies that are scheduled regularly during the acquisition life-cycle. Though monitoring is acknowledged to be extremely important, it has usually not been performed in the past. Lacking encouragement, it has been all too easy to go for months and even years without updates, or more important, verification of the initial sizing and timing estimates.

The procedures specified for each type of sizing and timing study are discussed in detail, step-by-step, in the remainder of this section. The discussions include approaches from Gilbert, et al. 7/

5.2 Conducting Initial Sizing and Timing Studies.

Procedures for conducting initial sizing and timing studies are presented in Figure 8 and are discussed in Sections 5.2.1 through 5.2.10.

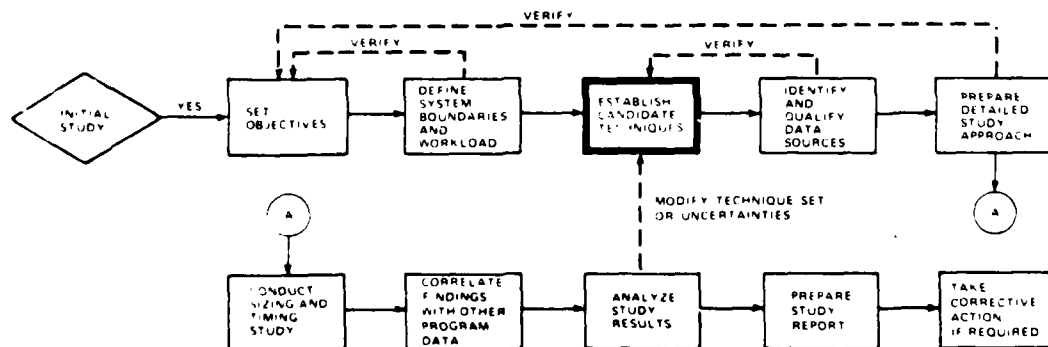


Figure 8. Sizing and Timing Initial Study Procedures

5.2.1 Set Objectives. The objectives of a study should identify specific needs such as initial resource estimates, comparison of alternative concepts, proposal evaluation, etc.

When structuring the objective the following should be considered:

- Specify the parameter(s) to be measured in precise terms, such as determination of main storage given 80% utilization at the workload specified, not just core required.
- Identify the resources available to conduct the study, in terms of personnel, funds, and time.
- Provide definitions of any terms to be used that are not standard, or that may be subject to interpretation.
- Specify the anticipated degree of accuracy of the final study result(s).
- Identify other organizations and individuals who will make use of the study results.
- Do not specify an obviously unrealistic objective.
- Do not plan to consume more resources than would be the cost penalty of not doing the study at all.

5.2.2 Define System Boundaries and Workload. The definition of the system or subsystem workload is the most important step in a sizing and timing study. It may not always be possible to accurately define the workload, especially during initial sizing and timing studies. In this case a notation should be made to this effect for historical purposes. In bounding the system and defining workload:

- Draw or identify a block diagram of the portion of the embedded computer system under evaluation.
- Specify all known or assumed hardware parameters. If the parameters of a hardware unit are the object of the study, identify the limits of the parameter set.
- Identify any known software characteristics that fall within the ECS boundary. This may include firmware (see definition in Glossary) considerations.

- Identify any software size being estimated as accurately as possible. It may be no more than simple application, but be sure to define what is meant. For example, if the software is for controlling Input/Output (I/O) devices, bound it by defining what devices will be controlled. Remember to include software for training and testing, if applicable.
- Define as many workload parameters as possible. A complete definition will include:
 - Job definition - A job is an action by the ECS that places a demand on system resources. This can be as simple as reading in two values from disc, adding them in core and printing one number, or it can be as complex as executing a subroutine with thousands of executable paths in a distributed system.
 - Job sequencing - Identify the sequence, or the probability distribution of job calls.
 - Job run time - Determine the time required for a job to run. This can be a discrete value for a sequential job, or expressed as a probability if, for example, there exists an 80% probability that a job will execute in a certain manner, then the execution time of the path may be calculated.
 - System resources required - Identify system components, such as terminals, discs, core, tapes, etc.
 - Job priority - Determine the priorities of jobs in a workload.
- Define any related workloads, such as that of a system being replaced.
- Identify assumptions - This is important in that often assumptions lose the quality of an unknown and assume a measure of fact if they are not verified and refined. Examples of assumptions that may have to be made are:
 - future trends in performance characteristics of hardware that may occur,
 - any workload parameter, such as job-arrival time that is not accurately defined, and

- basic system configuration that is not accurately described.

Once the system boundaries and workloads have been established, care must be taken to insure that the objectives of the study can still be met. Even if there are numerous assumptions, as long as they are accounted for, the study will have meaning.

5.2.3 Establish Candidate Techniques. The establishment of candidate techniques (see Figure 6) is a function of how much is known about the system under study, the study objectives, and how many assumptions may be made and still provide a useful product. It must be kept in mind that until the ECS is fully developed with all code and data in place, the size is uncertain. Also, until the system is exercised under actual operating conditions, the true timing is uncertain. In conducting a sizing and timing study there may be portions of the system that are fully developed or off-the-shelf, and therefore, some of the direct-measurement techniques may apply. In the same study, however, the logical flows of certain paths may be undefined and require the application of less precise techniques. The point to bear in mind is that all classes of techniques, i.e. extrapolation, representation, and measurement, should be considered in the context of objectives and data availability.

When selecting the candidate technique(s):

- Refer to Tab A and start with the most accurate technique category, such as monitors, and work backwards towards analogy.
- As a first cut at selection, be more concerned whether the required data can be obtained, not how it will be obtained or how much it will cost. This will result in a feasible set of techniques, but not necessarily a viable set.
- Understand the basic assumption required for the techniques use. For example, to use any Markovian technique 11/, it must be assumed that if a sequence

of jobs is to be modeled, the next step in the model is dependent only on the current state, and not upon any previous history. In some cases this may be a reasonable assumption, but in others it will not.

- Consider using multiple techniques to obtain estimates of various sizing and timing parameters.
- Be aware that by ignoring some parts of the bounded system that may be very complex, but with a low probability of occurrence, a simple estimate of the bulk of the system will be better than struggling with a very complex portion. On the other hand, a less frequently called part of a software program may be the most critical and, therefore, may have to be addressed at all costs. This will be clear if the study objectives and workloads have been clearly defined.
- Take advantage of other techniques that may be available to other ESD programs, such as simulator packages or measurement packages that may be under Air Force lease.
- Do not ignore current literature on the subject of performance monitoring. Since a system-engineering approach to the application of performance analysis to early estimates is in its infancy, new material will be constantly published. Take one or two days to review what is currently being done.
- DO NOT LOOK FOR MORE ACCURACY THAN IS POSSIBLE. AN ESTIMATE IS STILL AN ESTIMATE NO MATTER HOW IT IS MANIPULATED.

5.2.4 Identify and Qualify Data Sources. Once the candidate techniques have been selected, the sources of the data required to implement the technique must be verified. Each technique contained in Tab A indicates what types of data must be obtained or estimated in order to use the procedure. When identifying and quantifying data sources:

- Identify the specific source of the data, rather than use a phrase such as, "to be supplied by user." Identify WHO will provide it, WHEN it will be provided, and in what FORM it will be provided.

- Specify any unusual tolerance limits that a particular study may require. For example, early contractor software code counts are usually estimates of the total, and therefore, counts to the nearest 1000 may be sufficient. However, if trends indicate software growth, demand the latest, most accurate data. 1000 lines of code should represent 10 modules.
- Take advantage of other management systems that may be generating data that can be used. For example, the requirements of the Cost/Schedule Control Systems Criteria (C/SCSC) have been imposed, request the backup that provides the basis for calculating the Budgeted Cost of Work Performed (BCWP) in the area of concern.
- Evaluate the risk of not having certain data, but if a worst-case analysis indicates satisfactory margins still exist, the missing data is not critical to the study.
- Do not request or obtain more data than can be analyzed, either in terms of time or of the skill of the estimator, unless steps are being taken to acquire additional technical assistance.

5.2.5 Prepare a Detailed Study Approach. Once the objectives have been set, the system bounded, candidate techniques selected, and data sources identified, a detailed study approach must be developed. It should be remembered that the detailed approach can be anything from a single page to twenty pages or more, depending on the complexity and requirements of the estimate. Even for a small study, a detailed study plan should be prepared. This plan will form the first part of the study report. As a minimum, the plan should contain the topics listed in Table 2 below, and should be expanded as required.

Once the detailed study plan is written, it must be reviewed to insure that the original objectives will be met.

TABLE 2. TOPICS FOR STUDY APPROACH

1. Tasking authority
2. Objective
3. System boundaries
4. Constraints and assumptions
5. Resources assigned
6. Techniques that will be used
7. Sequence of activities
8. A predetermined table in which to enter results.

5.2.6 Conduct the Sizing and Timing Study. Conducting the sizing and timing study is in the same category as acquiring the system under study; if everything goes as planned there should be no problems, but things rarely do. The study team should do the tasks outlined in the detailed plan, based on the data and objectives. In the event that previously identified techniques and supporting data simply cannot be made to work, fall back to the next level of estimating techniques and continue. This will cause the anticipated error to be larger, but will still provide useful results. When conducting the study:

- Be alert for inconsistent data. Such problems discussed prior to the analysis of the study results will save time and resources in analysis.
- Keep an organized record of events during the study, for historical purposes.
- Keep an open dialog among all members of the procurement team and users or potential users of the system.
- Do not become diverted from the objectives of the study. If a sudden requirement emerges for additional estimates during the study, either begin again or modify the study accordingly. In any event, document it.

5.2.7 Correlate Findings With Other Program Data. This step can yield a high payoff in terms of total program success. There is a tremendous amount of data that is either scheduled to be received or is being

received during an acquisition. At the end of a program all the pieces should fit: total size, total resources consumed (including time), and total system performance. All these things should relate during development. If they do not, the potential for a problem exists. For example, if the study estimate results indicate that 500,000 lines of source code are required, the contractor estimates 100,000, and the development time could realistically, from historical data, support only 50,000, there exists a potentially critical problem.

When correlating sizing and timing estimates:

- Be aware of built-in errors in the results.
- Anticipate that most unsupported estimates are optimistic.
- Demand backup and rationale for any supporting data.
- Make sure you are relating similar units. Remember any measure of system timing is workload dependent.

5.2.8 Analyze Study Results. In analyzing the sizing and timing study, the participating personnel must keep in mind the following:

- The final results may contain three types of errors; 1) erroneous assumptions, 2) data errors, and 3) computational errors.
- Unless the study was based completely on measureable data in a live environment, the results are only an estimate and must be presented as such.
- Sensitivity analysis should be conducted to determine the effect of workload variation, equipment parameters, and estimated data.
- Do not let the analysis overpower the data. If the objective of the study is a gross estimate of core requirements, the analysis should correspond to that level of effort.

5.2.9 Prepare the Study Report. Preparing the sizing and timing study report is a key factor in the process, since management decisions will most likely be made based upon the results. In addition, the report will form the basis for data collection and as the source document for updated studies with comparable objectives. Some guidelines for the preparation of the report are:

- Keep it as brief as possible.
- Bound the report to the stated objective. Any information or problem discovered that is worthy of an additional study, or is important to the program, but beyond the scope of sizing and timing, should be put in a separate appendix and brought to the attention of appropriate personnel.
- Insure that terms are fully described where confusion could result when later using the report.
- Clearly spell out limitations in the use of the data.
- If the study is a failure, say so and indicate why and how, and when it will be redone.
- Structure the report to cover the topics listed in Table 3 below, at a minimum.

TABLE 3. TOPICS FOR STUDY REPORT

- | |
|---|
| <ol style="list-style-type: none">1. Introduction2. Tasking Authority3. Objective(s)4. Constraints and Assumptions5. Bounded System6. Workload7. Physical Component Characteristics8. Techniques Used9. Conducting the Study10. Analysis of Results11. Assessment of Accuracy12. Other Program Implications13. Recommendations for the next study, and corrective actions to be taken, if required. |
|---|

5.2.10 Corrective Action. Although corrective action is not a function of the sizing and timing study personnel per se, it is necessary that a follow-through exists whereby these individuals (it may be those who conducted the study, but with a different title) who must take action are personally made aware of the report. The following guidance is provided in the area of corrective actions:

- Involve all who contributed to the report in the proposed corrective actions.
- Involve the developing contractor, appropriate Program Office personnel, and the user in the proposed corrective action so that in the event the report is incorrect, discussions can take place.
- Use the report as a positive tool; most contractors want to be successful, give them a chance.
- Document any corrective actions taken and append them to the report. This will assist future sizing and timing study personnel in their job.
- Any corrective actions that may have a contractual impact must be coordinated with the Contracting Officer.

All of the above steps in conducting the sizing and timing studies may and should be modified to meet the objectives. However, by structuring the logic of the study approach, it will be possible to improve the methodology of conducting estimates.

5.3 Study Update.

Procedures for conducting sizing and timing study updates are presented in Figure 9 and are discussed in Sections 5.3.1 through 5.3.7.

Updating a sizing and timing study must be conducted with two goals in mind. First, there must be a striving for an improved estimate of the parameter under study, whether or not the improved estimate is moving in the desired direction (i.e., reduced response time), and second, the amount of detail that has become hard fact between estimates should be

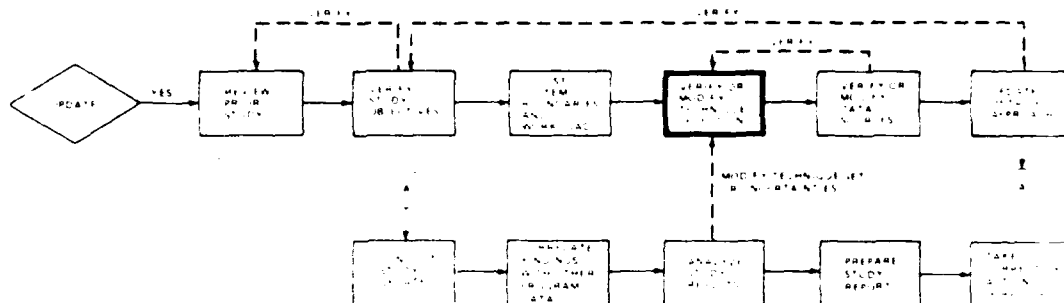


Figure 9. Sizing and Timing Update Procedures

Figure 9.

increasing. One danger in the development of C3 systems with embedded computer systems is that long periods of time may pass with no measureable improvement in sizing and timing parameters. Initial contractor estimates that indicate no problems exist, are likely to be accepted without question until a time when an event that was predicted to go smoothly--such as subsystem test--begins to indicate slow response time.

Many of the steps in updating the sizing and timing study are the same as for the initial study, but the differences in some steps are significant. The following paragraphs discuss the differences in procedures (see Figure 7).

5.3.1 Review Prior Study. When the determination has been made to update a sizing and timing study, the first step is to review the previous study in order to verify that what is required is truly an update and not a new study. There can be no hard and fast rules that separate the update from an initial study, however the below listed points should be considered:

- UPDATE IF:
- The parameters being estimated are the same as a previous study.
- and
- There existed a degree of uncertainty in the previous study that would result in exceeding safety margins (in terms of storage and timing).

- and
- There have been no major changes in the configuration of the ECS.

5.3.2 Verify Study Objectives. The objectives for a study update are normally to refine a previous study. There may be a tradeoff between increased system definition, that would reduce the anticipated error, and an expansion of the objective scope, that may increase the error. The requirement is that an audit trail be maintained between the two studies.

5.3.3 Adjust System Boundaries and Workload. The boundaries of the previous system may be expanded, or reduced, as required to support the new objectives. There is a distinction here between system configuration and system boundary from the study viewpoint. The boundary is the subset of the ECS that has been chosen to support the objective. The verification of the workload must also be adjusted as necessary to support the updated requirements of the system. Multiple workload scenarios may be included as the system definition progresses throughout the acquisition. The guidance provided in paragraph 5.2.2 should be used in verification for the update.

5.3.4 Verify and/or Modify Technique(s) Selection. This is the area that requires an orderly transition from the classes of techniques that are used with limited data (Extrapolation) towards more accurate techniques (Measurement) (see Figure 6). The difficulty arises when long periods of time are allowed to transpire with no measureable improvement in system definition or construction. In a long development program; however, this may be a fact of life, but there must be a concerted effort to always look ahead and predict when the level of knowledge will support a study update. As is presented in Section 5.4, Monitoring, the reviewing of data provided by the contractor under the Contract Data Requirements List (CDRL) can be of assistance in determining when refined techniques may be employed.

5.3.5 Verify and/or Modify Data Sources. The data sources used in the previous study should be consulted for applicability to the study update. Sources that proved to be invalid or not available should be reviewed as to whether they have improved or whether they should be deleted. New data sources, which will support improved techniques, will become available as the system progresses. Care must be taken, however, as in the initial study, to carefully weigh the reliability of data sources. This may require the Program Office to request backup data to support any changes, (or no change) in contractor estimates of code counts and execution times. The selection of data sources must be fed back to insure that they will support the techniques selected.

5.3.6 Update Detailed-Study Approach. The previous detailed-study approach should be reviewed in two specific areas. First, does the updated approach support the new study objectives, and second, should the steps taken in the previous approach be modified or eliminated due to updated requirements or lack of usefulness to the earlier study. It is expected that as the skills of the estimating personnel increase, the detailed-study approach will become more effective.

5.3.7 Conduct Study Update. Based on the previous steps, the estimating personnel conduct the study, hopefully more efficiently. Detailed record keeping of the conduct of the study update is even more important than when conducting the initial study, because the level of detail of the study will be increasing, and it becomes more difficult to correct errors further into the program. In addition, the history of studies will provide the basis upon which to build the overall skills of ESD personnel involved in the estimating process.

When the study is complete, as indicated in Figure 7, the steps then follow those specified for the conduct of the initial studies as described in paragraphs 5.2.7 through 5.2.11.

5.4 Developmental Monitoring.

Procedures for conducting sizing and timing developmental monitoring are presented in Figure 10 and are discussed in Sections 5.4.1 through 5.4.8.

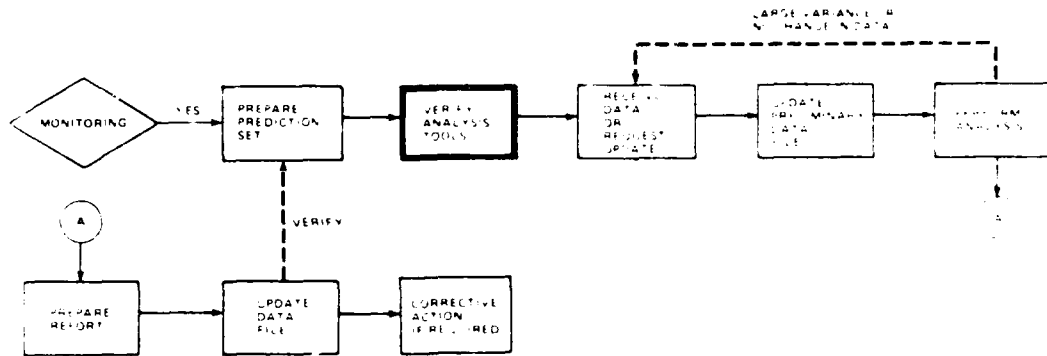


Figure 10. Sizing and Timing Developmental Monitoring Procedures

There has been a tremendous amount of material written and studied, over the past few years, on monitoring the development of weapon systems. The primary management system used throughout the Department of Defense is the Cost/Schedule Control Systems Criteria (C/SCSC) promulgated by Department of Defense Instruction (DODI) 7000.2. This system is in fact a set of criteria to be followed, and could be imposed without receiving any data. Data in support of C/SCSC must be requested on the Contract Data Requirements List (CDRL), usually in the form of the Cost Performance Report (CPR). This system, however, is based upon dollars and manmonths as the reporting variables, and all technical achievement is translated into these terms. There also is a great deal of latitude in extending the Contract Work Breakdown Structure (CWBS) for a given acquisition. Although MIL-STD 881A is to be used for the project and contract WBS, the extended WBS is normally developed by the contractor and approved by the procuring agency.

The efforts to date in the acquisition of data for C³ systems, and in fact all major systems that have embedded computer systems, have been

directed at developing resource measurements in terms of manmonths, computer time, and development time. No significant effort has been made to collect functional system data on sizing and timing, so therefore, no adequate sizing and timing data bases exist that can be used by Program Offices or ESD/TOI. Obviously the measure of manmonths is unique and well understood; however, system size and measures of timing are not agreed upon, or even defined in a minimum number of ways. For example, a single numerical timing value for a system is practically meaningless without the specific description of the system's workload and the ECS configuration.

There are five primary factors that have contributed to the lack of ECS developmental monitoring:

- It is difficult.
- The data is not available in early stages.
- Data collection forms are inadequate.
- C3 systems are unique, hence workloads vary widely.
- It is time consuming.

Developmental monitoring as shown in Figure 10 is started with the receipt of the first routine Data Item Description (DID) report submission containing sizing and timing data. There is a wide range of material, from simple progress reports, to the C-5 specification, and all material in between. In order to bound this portion of this report, one must consider only material that can be cast in the form of a DID dedicated to sizing and timing data. The personnel involved in sizing and timing estimates, however, must use technical data in all forms, particularly with respect to the step "Correlate Findings" presented in paragraph 5.2.7 above. All levels of specifications include information that must be used in conducting sizing and timing studies, as well as such DID generated documents as:

- Computer Program Development Plan,
- System (Segment) Specifications,
- Addendum Specifications,
- Product Specifications,
- Technical Report (Sizing and Timing Data),
- Computer Program Detail Specification,

- Computer Program Flow Charts, and
- Computer Program Sizing and Timing Data.

There are many more DIDs that generate sizing and timing data. In fact, part of the problem that has resulted from the lack of success in estimating and monitoring sizing and timing parameters has been the excess of data without any coherent order. Basically, the lack of monitoring on a routine basis is more of a contributor to disasters in the area of ECS sizing and timing than any other factor once requirements have been established. It is important to stress that the lack of monitoring is even more critical than poor initial estimates. It is a fact of life that initial estimates are only that. As such, initial estimates can not, and should not, be expected to be precise. This is not to say that improvement can not be made, but part of the improvement is the acceptance that extensive design efforts must be done before the quality of a given estimate is even within 50% in the case of core requirements.

5.4.1 Prepare Prediction Set. The first step, as indicated in Figure 10 for developmental monitoring is the establishment of a Prediction Set form. The entire process of the seven steps shown should be completed every month, with the eighth step as soon as possible. This is not only a natural measure that coincides with other program schedules, but relates to the delivery of financial reports. A dedicated individual or group should be assigned the responsibility for sizing and timing monitoring. The Prediction Set will be a group of data collection forms each with a specific, measureable value, supported by constraints. An example of a Prediction Set form is presented in Figure 11.

The system or subsystem configuration, workload, and other constraints should be specified in a separate section of a monitoring file, maintained in either manual or automated form, and always retained for an audit trail. When all key values are established, a master record should be constructed to provide for the bounded sizing and timing parameters being tracked at a given time.

PREDICTION SET Month <u>16</u> PROGRAM: <u>KEY WATCH</u> DATE: <u>22 Feb. 1980</u> EVALUATOR: <u>Lt. R. B. Jones, USAF</u>		
1	KEY VALUE (and Parameters)	Main Memory Utilization (64K - 32 Bit/Words)
2	LAST VALUE Prediction Set Month: <u>15</u> of: <u>25 Jan. 1980</u>	Estimated Value on Completion 57% Utilization (36.5K)
3	PREDICTED VALUE AS OF THIS DATE	Estimated Value on Completion 60% Utilization (38.4K)
4	RATIONALE FOR PREDICTION	Estimated increase in application software required to produce two new reports R-6 and R-7
5	CONSTRAINTS AS OF THIS DATE	System Configuration: No Change from Prediction Set Month 3 Workload: Same as Prediction Set Month 5 plus new reports R-6 and R-7
6	VALUE ESTIMATED THIS MONITORING PERIOD	59% utilized (37.8K)
7	COMMENTS (If Significant Difference Between Items 3 and 6)	None
8	OTHER COMMENTS	Item 6 value based on Contractor's estimate of increased lines of source code needed for reports R-6 and R-7 (see analysis in this month's monitoring report)

Figure 11. Example of a Completed Prediction Set Form

5.4.2 Verify Analysis Techniques. For conducting monitoring of sizing and timing estimates, the same set of techniques from Tab A are used. Some of the key factors in technique selection are:

- Try to apply as many techniques as possible, even if large errors may exist. Select techniques that will produce more accurate results if data and funding are available.
- Stress the evaluation of incremental work as it relates to increased definition and risk reduction. That is, probe to find out how the contractor spent his time, because you may be sure he spent money.
- Have the techniques selected, set up, or automated so that you may begin analysis as soon as the monthly data arrives.

5.4.3 Receive Data or Request Update. On any given program, close dialog must exist with the Program Office's data manager to evaluate what data items contain sizing and timing data. Some steps to take prior to and after the receipt of data are:

- Identify the documents to be reviewed.
- Identify sizing and timing data anticipated from each.
- Develop a matrix to identify similar items from different data sources.
- Identify other Program Office technical personnel who will be using the data.
- When data are received, note the arrival and verify that the sizing and timing information is present; or if not, determine why not.

5.4.4 Update Preliminary Data File. All data that affect sizing and timing should be annotated and transferred to the Prediction Set form as appropriate (see Figure 11). The next step is to perform required analysis. It may be that no analysis is required in a particular area because the previously estimated value is still valid, such as the

firmware in a microprocessor. What may happen at this point is that instead of estimating the size of firmware, the emphasis may shift to the performance of the firmware.

5.4.5 Perform Analysis. The performance of sizing and timing monitoring analysis is really a sizing and timing study, but on a smaller scale. Therefore, the steps from the verification of the objective, through the start of report preparation, are the same. For example, the item being studied could be the monitoring of the coding of 5 modules, or about 500 lines of code. It has been estimated that a programmer can normally produce about 100-150 lines of complex code a month. In the event data indicates that only 50 lines of code were delivered, it may indicate the programmer is encountering difficulties and the initial sizing estimate for the coding of the 5 modules was underestimated. The analyst must always be alert for the indication of excessive delays in coding, that may translate into software size overruns. It also may be that the programmer coded 300 lines, and threw away 250. Excessive code breakage is another precursor of software size growth.

5.4.6 Prepare Report. The report prepared as a result of routine monitoring provides three things:

- It forces routine reviews of selective sizing and timing estimates.
- It provides an historical data base (although unstructured).
- It provides the basis for corrective action, either low level or the initiation of a more detailed sizing and timing study.

The report should be kept brief, and structured to be most beneficial to the Program Office. The routine monitoring report should be simple, and specific to the acquisition. For that reason no elaborate format is presented herein.

5.4.7 Update of Data File. When the monthly report is complete, each of the values estimated should be entered as the VALUE ESTIMATED THIS MONITORING PERIOD on the Prediction Set forms (see Figure 11) in the program's data file. This will close the loop for a given month and prepare the analyst for the next cycle. It may well be that not all parameters can or need be analyzed every month. This will be determined by the Program Office's capability, size of the acquisition, amount of changes impacting on sizing and timing parameters, and the risks involved. However, some analysis in the areas of sizing and timing should be conducted each month.

5.4.8 Take Corrective Action. If it appears that corrective action should be taken, refer to Section 5.2.10.

6. INTEGRATION OF SIZING AND TIMING

The increased emphasis that must be placed on sizing and timing has many facets woven throughout the acquisition life-cycle. All areas of systems engineering have aspects that may be affected by inadequate system operation caused by the breaching of sizing and timing estimates. Three important aspects of sizing and timing are: the discipline of sizing and timing; the establishment and retention of spare memory capacity; and sizing and timing data collection.

6.1 Sizing and Timing as a Discipline.

The discipline of sizing and timing estimation must be fit in the overall scheme of system acquisition. There exists a wide body of technical knowledge in two general areas, Computer Performance Evaluation, and Computer Systems Performance Measurement. For years, techniques such as simulating, modeling, benchmarking, and monitoring have been used, but primarily in the environment of a data processing center operating in a batch or sequential-job mode. Usually these techniques were applied to optimize or fine tune an existing system. With the recent explosion of time-sharing systems and distributed processing, however, much more emphasis is being placed on systems that are becoming more real-time, like C³ systems. A recent example of this increased emphasis may be seen in the papers presented at the Conference on Simulation, Measurement and Modeling of Computer Systems, held in Boulder, Colorado, August 13-15, 1979. 16/

Since estimating sizing and timing parameters is really a continuous process for insuring that a system performs effectively, one can recognize the association with Verification and Validation (V&V) activities. Two of the Software Acquisition Management Guidebooks 6/, 18/, that discuss verification and validation of software, also address sizing and timing at the system level. This recognizes the interrelationships between hardware, software, firmware, and workload. For example, when one is attempting to estimate the size of the object code for a system, based

on data regarding the estimated size of the source code, one must know the source to object code conversion factor for the compiler. In certain cases where storage capacity might be critical, one might consider the relaxation of a specified High Level Language (HLL) for the utilization of one with a more efficient conversion factor. Furthermore, the sizes of core memory and necessary resident software and data may dictate a need for overlays. Such a need can result in increased software size and reduced processing time, not to mention the probable impact on future modifications. These interrelationships should be kept in mind and tradeoff analyses conducted when appropriate during the Conceptual Phase. (Also see AFR 800-14, Volume II, paragraph 3-4b.)

6.2 Spare Memory Capacity.

One sizing parameter that might seem easy to obtain, and yet is one of the most difficult parameters of the system to define, is spare memory capacity. In some instances this has been defined as no more than a percent of main memory (in thousands of object words) that must be reserved for future growth. This definition is inadequate, for it does not consider mass memory requirements.

Herd, et al 9/ developed an algorithm that demonstrates a relationship between core utilization and a resource multiplier. This relationship illustrates that increased core utilization results in increased cost to develop a given amount of software. From core utilization of less than 60%, expanding to 80%, a cost increase of up to 30% per instruction for the total program can be expected, and beyond 80%, cost increases of up to 200% have been noted. Therefore, it becomes critical to specify not only memory reserve, but also the reserves of all resources. A reserve of 20% is considered the absolute minimum and 40% reserve would be much more appropriate. Although AFR 800-14 Volume II, paragraph 3-4b states that reserve storage capacity should be available, it does not specify the amount. MIL-STD 1679 (Navy) does state that the total system memory capacity should contain at least 20 percent reserve capacity, at the time of acceptance.

Acknowledging that there will probably be growth in resource requirements due to requirements changes and misestimates, it is recommended that initial reserves in these categories be not less than 40% entering the Full-Scale Development. Once again it is imperative to specify under what conditions this minimum reserve capacity will be measured, in terms of workload. Close coordination must exist among the personnel involved in the generation of test plans and procedures to insure that no misunderstanding exists.

The ESD/TOI guidance for estimating total memory utilization in order to determine the reserve total memory available is provided in the following equation. The reserve should equal one half of the memory capacity necessary for all the data and programs required for a system to operate effectively:

$$D + P + 1/2(D + P) = T_M \quad (\text{Equation 1})$$

where

- D = Total number of object words (in thousands) of data.
- P = Total number of object words (in thousands) in system programs.
- $1/2(D + P)$ = Reserve memory capacity (in this case equal to 33 percent of T_M).
- T_M = Total number of object words (in thousands) (this should equal the specified capacity of the total memory).

Although the above equation is for estimating total memory capacity or reserve total memory, it should also be used to determine the total core or main memory capacity. In this case, the total data, programs or portions thereof, that must be resident in the main memory at one time, should be used in the equation.

6.3 Sizing and Timing Data Collection.

As stated in Section 5.4, the monitoring of sizing and timing estimates is a serious problem. In reviewing Data Item Description (DID)

DI-S-30568, Computer Program Timing and Sizing Data, and its application to ESD projects (see Volume II, Tab A1 for more information on ESD projects), two points were determined:

- DI-S-30568 is not routinely applied to ESD acquisitions.
- DI-S-30568 is not sufficient for the routine monitoring of sizing and timing estimates.

It is possible that DI-S-30568 is not being routinely applied because of the statement in paragraph 1. of the DID, that states, "However, this DID shall not unnecessarily duplicate descriptive material presented in other documents." There could exist a question as to the meaning of "unnecessarily" and "descriptive material" as opposed to technical data. In addition, no format for data display is provided, and more importantly, no specific mention is made of workload. Also there should exist an audit trail from page 9 of the ESD Computer Program Development Plan (CPDP) backup sheet to any routine reporting of sizing and timing parameters. Any DID used to specify input documents must be a viable document that encompasses the entire system acquisition life-cycle process. During the Conceptual Phase, initial sizing and timing estimates and reserves should be recorded by whatever activity is involved, and these estimates, and subsequent actual data, should be retained through Full-Scale Development (FSD). The data will change and possibly bear no relationship to initial estimates, but an audit trail would be maintained if the procedures in this Handbook are used.

TAB A INTRODUCTION TO SIZING AND TIMING TECHNIQUES

1. INTRODUCTION

1.1 Considerations for Technique Selection.

The selection of techniques for use in estimating the sizing and timing parameters of a computer system, should be a function of three considerations:

- the amount of information known about the system or subsystem to be studied,
- the sizing and/or timing study objectives, and
- the number of assumptions that can be made regarding the system or subsystem and still produce a reasonable estimate.

1.2 Technique Applications.

Application of techniques are discussed in Sections 4.3., 5.2.3, 5.3.4, and 5.4.2. Eighteen major techniques suggested for use are presented in Figure A-1 below and are discussed in more detail in this Tab. (See section 1.3 below for the index of techniques). Use of these techniques requires a great deal of special experience and education. In the event ESD personnel need assistance in conducting a sizing and timing study, contact ESD/TOI. Note that as techniques move from extrapolation to actual measurement, the expected reliability of each category of techniques is higher. There is no sharp distinction among three of the categories, analytical, models, and simulations, but rather, the primary difference is in the techniques and the subsequent reliability of the results. Any specific technique may be combined with another; for example, models may combine several analytical techniques, or a deterministic model may include empirical equations and queuing theory equations, or else a simulation may include both deterministic and probabilistic models. Furthermore, there are times when the use of more than one technique is appropriate in estimating different sizing and timing parameters of a system. For instances, one might conduct analogies

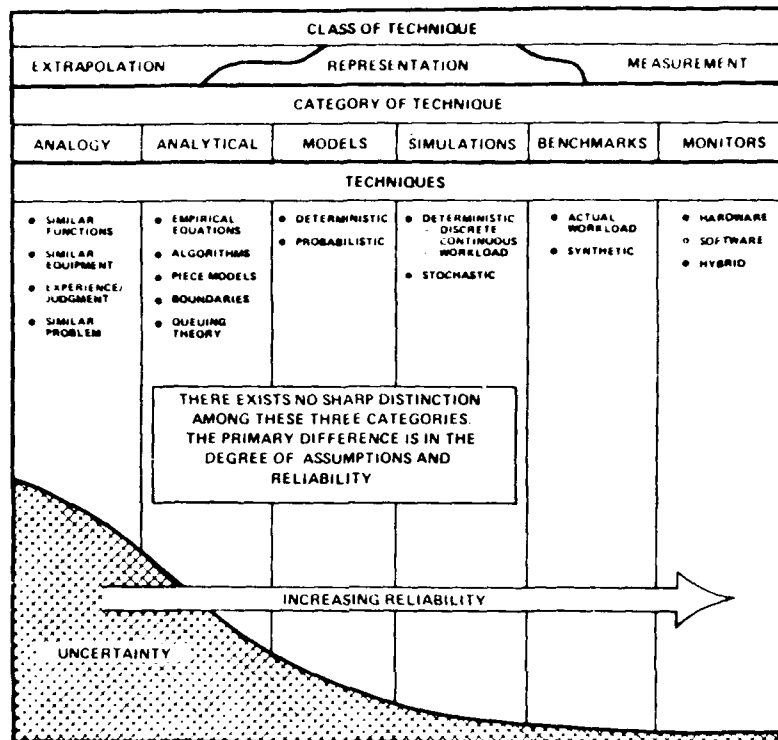


Figure A-1. Sizing and Timing Techniques

by comparing similar functions and equipments to obtain initial sizing estimates for core or total memory requirements, and have simulations performed to estimate the throughput and response times of certain subsystems or the entire system.

1.2.1 Analogies. Though techniques categorized as analogies are the least reliable methods of estimating computer system sizing and timing parameters, they are often the only techniques that can be applied in the early phases of the system acquisition due to the lack of data regarding the system definition. In making an analogy, the more data one has

regarding both the new system and the system to which it is being compared, the better the estimate. In using an analogy, the estimator may err in the estimation of the requirements for the new system by not considering:

- technology advancements that may have occurred since the development of the old system,
- inadequacies that exist in the old system, such as lack of reserve memory capacity or software discrepancies that required modification during the deployment phase,
- inadequacies in the new system specifications that will result in changes, and
- differences between the two systems that will impact on the evaluation.

1.2.2 Analytical Techniques, Models, and Simulations. These categories of techniques basically use the same tools and only differ in the degree of complexity. The Analytical group may be considered appropriate for rough-cut analysis of a general system description from the requirements rather than any specific system. Use of the Models group might be considered appropriate when there is a good system definition and one or more hardware systems need evaluation. The Simulations group performs the imitative functions of a system based on the best knowledge of the required functions and workload at the time.

1.2.3 Benchmarks. The Benchmarks group may use the actual or similar hardware, and may employ an actual or synthetic workload. The synthetic workload may be based on assumed characteristics of the software, and such other assumptions as the amount of processing required to complete certain functions or jobs, or the amount of data to be processed. Actual workload benchmarks use the actual software under assumed operating conditions and are considered more reliable than the synthetic.

1.2.4 Monitors. This technique category is primarily used to evaluate performance of completed or nearly completed operational systems. A hardware monitor collects data on system usage, utilization, idle time, etc., and is transparent to the user, in that it does not occupy memory, nor does it impact on the timing of the system. Data from such a monitor may be used for sizing and timing studies, and also optimizing systems. In the event a system is approaching saturation or 100 percent utilization, a hardware monitor will not indicate the amount of delay due to overloading. The software monitor provides better indicators of particular functions than does the hardware monitor. The overall condition of a system may be estimated to be excellent through the analysis of hardware monitoring data; however, software monitoring data may indicate that certain critical functions are not performed within prescribed tolerances for specified operating conditions. With a software monitor, the size of the queues can be recorded and the extent of saturation identified to a greater degree than with a hardware monitor. A disadvantage of software monitors is that they have a sizing and timing impact on the systems they are monitoring and accordingly, the monitoring data is somewhat distorted. A hybrid monitor is a combination of both hardware and software monitors, and has the advantages of both. The hybrid monitoring data is more complete than either monitor operating separately, though it is less reliable than the hardware monitor due to the impact of the software monitor portion of the hybrid.

1.3 Tab A Index of Techniques.

Sizing and Timing techniques are discussed in more detail on the following pages of this tab.

<u>Category of Technique</u>	<u>Technique</u>	<u>Page</u>
Analogy	Similar Functions	A- 7
	Similar Equipment	A- 9
	Experience and Judgment	A-11
	Similar Problem	A-12

<u>Category of Technique</u>	<u>Technique</u>	<u>Page</u>
Analytical	Empirical Equations	A-13
	Algorithms	A-14
	Piece Models	A-15
	Boundaries	A-17
	Queuing Theory	A-18
Models	Deterministic	A-19
	Probabilistic	A-20
Simulations	Deterministic	A-21
	Stochastic	A-22
Benchmarks	Actual Workload	A-23
	Synthetic	A-24
Monitors	Hardware	A-25
	Software	A-26
	Hybrid	A-27

CATEGORY OF TECHNIQUE: Analogy

TECHNIQUE: Similar Functions

DESCRIPTION: This technique assumes that a person can estimate the sizing and timing parameters of a proposed computer system by comparing identified functional requirements of the proposed system to those of an existing computer system. This technique may be approached from two directions: the top-down approach starting with estimates based on a similar system used in a similar application; or the bottom-up approach, breaking the proposed system into individual functions and summing the results. In the top-down approach, the estimates are refined as more data about the system becomes available. In the bottom-up approach, functions may be broken into subfunctions to obtain more refined estimates.

REQUIRED DATA: Factors that should be considered, depending on the level of system definition, are listed in Table 1. For the comparative system in a top-down approach, the first consideration should be the system application. See Tab A in Volume II of this Handbook for examples of recent ESD systems. Next, specific computer equipment and their characteristics should be considered. This should be followed by discussions with contractors, users and Program Office personnel (if the office is still in existence) to obtain more specifics, such as differences in functions between the two systems or the amount of actual reserve memory capacity. Although the above discusses comparing two systems, it does not preclude comparing the proposed system to more than one system or a number of subsystems. Multiple comparisons are encouraged and should increase the levels of confidence in the developed estimates.

ASSUMPTIONS: The assumptions for both top-down and bottom-up approaches are similar. Basically, each approach assumes that systems that perform

ASSUMPTIONS:
(continued)

similar functions are similar in other aspects, such as workload. It is further assumed that differences due to expended functions, improved technology, or system configurations can be accounted for and estimates adjusted accordingly.

APPLICATION:

Initial estimates of similar functions may be in gross figures; however, as more data becomes available by system, subsystem or function, the estimates should be refined. In both the top-down and the bottom-up approaches, a point should be reached that produces relatively identical results.

LEVEL OF CONFIDENCE:

One's level of confidence is very dependent on the quality of data available and the extent to which the data is analyzed. Normally, the more complex the systems are, the greater the probability that inaccurate comparisons will be made, and accordingly, one's level of confidence will decrease.

CATEGORY OF TECHNIQUE: Analogy

TECHNIQUE: Similar Equipment

DESCRIPTION: This technique assumes that a person can estimate the sizing and timing parameters of a proposed computer system by comparing like systems, while assuming that the applications of the systems are similar.

REQUIRED DATA: Data pertaining to the hardware of a system being developed or in existence, (similar to the information presented in Tab A of Volume II of this Handbook) should be collected and compared with similar data on the proposed system hardware. Not only should equipment be compared, but also the configurations of the systems, since configuration differences may impact on the timing parameters.

ASSUMPTIONS: To assume that comparing similar equipment will result in a good correlation of sizing and timing parameters, one must also assume that the equipment will support similar applications and workloads. Assumptions regarding improved technology must be considered and estimates appropriately revised.

APPLICATION: First define the system to be estimated. Next, examine and evaluate similar configurations and chose one or more systems or subsystems that approximate the application being developed. Given special consideration to workload and job flow relationships. Evaluate impacts of configuration differences, technology changes, workload differences, as well as any other constraints that can be identified. Perform evaluation to determine sizing and timing estimates.

LEVEL OF CONFIDENCE:

The level of confidence in the results of the application of this technique is very dependent on the quality of data available and the extent to which the data is analyzed. The more complex the systems are, the greater the probability for errors, with a resulting decrease in one's level of confidence.

CATEGORY OF TECHNIQUE: Analogy

TECHNIQUE: Experience and Judgment

DESCRIPTION: This technique assumes that a person can estimate the sizing and timing parameters of a proposed computer system by applying one's experience with similar systems or subsystems to more readily determine what is needed to meet the requirements of a proposed system.

REQUIRED DATA: Since the basis of this technique is experience, the data is basically one's experience and capability to recall specifics of systems or subsystems with which one is familiar, or know where appropriate sizing and timing data may be obtained.

ASSUMPTIONS: It is assumed that one's exposure to similar systems or subsystems is adequate and reliable. The results are dependent on judgment.

APPLICATION: The application of this technique relies on one's experience, and the ability to recall details of systems or subsystems one is familiar with and relate the knowledge to the proposed system. Based on judgment, one must select those systems or subsystems that closely approximate the requirements of the proposed system. Next obtain and evaluate sizing and timing data of the systems being compared and estimate sizing and timing parameters.

LEVEL OF CONFIDENCE: The quality of a person's knowledge retention and one's ability to recall or obtain data determines the level of one's confidence in the results. Basically, if the person making the comparison is the one responsible for the results, his or her level of confidence should be higher than if another person is responsible for the results and does not have the experience.

CATEGORY OF TECHNIQUE: Analogy

TECHNIQUE: Similar Problem

DESCRIPTION: This technique assumes that a person can estimate the sizing and timing parameters of a proposed computer system by comparing similar system applications without singling out specific software or hardware.

REQUIRED DATA: Collection of sizing and timing data pertaining to similar computer system applications.

ASSUMPTIONS: It is assumed that the computer systems or subsystems being compared are similar in application, in many respects, and that differences can be identified and evaluated to determine their impact on sizing and timing parameters.

APPLICATION: After collection of sizing and timing data, reevaluate assumptions regarding applications, and refine parameters. Evaluate identified constraints on each application and determine relative impacts. Especially consider the relevance of differences in workload and, if data is available, differences in job flow. Finally, conduct an overall evaluation to determine sizing and timing estimates.

LEVEL OF CONFIDENCE: The level of confidence depends on quality of data and the accuracy of the predictions of differences and their relative impact on the systems being compared.

CATEGORY OF TECHNIQUE: Analytical

TECHNIQUE: Empirical Equations

DESCRIPTION: Empirical equations are based on the analysis of empirical data obtained by observation or experience. The development and use of equations containing variables that describe certain characteristics of a proposed computer system, in order to estimate its sizing and timing parameters, are based on data developed through simulations or actual systems. The equations may range in scope from predicting the characteristic of one function of a system to predicting the overall characteristics of an entire system, though the latter is highly unlikely at this current state-of-the-art.

REQUIRED DATA: The equations must be based on data that have common characteristics, to the maximum extent possible, with the system, or certain aspects of the system, examined. Particular attention should be directed towards workload and job flow characteristics.

ASSUMPTIONS: It is assumed that the empirical equations are based on data similar to the system being evaluated.

APPLICATION: Either develop or obtain equations that use characteristics similar to the system or portions of the system being evaluated. If the equations are obtained, variables in the equations may require revision to relate to the system being evaluated. Once the equations are applied, examine the results. If possible, use the equations with data from actual systems to determine accuracy of the equations and revised variables.

LEVEL OF CONFIDENCE: One's level of confidence will depend on the results obtained from the various equations. As one becomes more familiar with certain equations, confidence may increase or decrease.

CATEGORY OF TECHNIQUE: Analytical

TECHNIQUE: Algorithms

DESCRIPTION: The development and use of algorithms employ procedures that require iterative algebraic steps to obtain solutions. Also see discussion of algorithms in Tab B of Volume II of this Handbook.

REQUIRED DATA: The data required will depend on the specific algorithm or algorithms being applied. Use algorithms for which data is available.

ASSUMPTIONS: Specific assumptions depend on the algorithms developed or selected for use. Algorithm limitations must be strictly followed.

APPLICATION: Procedures of developed or selected algorithms must be followed. Data applied to the algorithms must be defined exactly as specified. Constraints should be observed and caution taken not to extrapolate results beyond allowable limits.

LEVEL OF CONFIDENCE: The level of confidence in an algorithm will depend on the accuracy of its development and the efficiency with which it is applied. One's confidence will depend on results obtained through use.

CATEGORY OF TECHNIQUE: Analytical

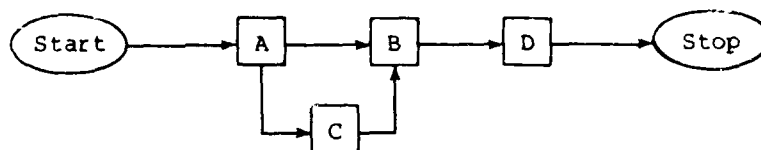
TECHNIQUE: Piece Models

DESCRIPTION: Piece models indicate performance of part of a system. Examples are modeling a buffer, or number of terminals to estimate average response time. A piece model could be the application of a model by Halstead 8/ that examines all operations in terms of operands and operators or the application of a Markov model 5/.

REQUIRED DATA: The data required depends on the specific model.

ASSUMPTIONS: In using a piece model to represent a portion of a system, one must make assumptions about the remainder of the system.

APPLICATION: A system can be partitioned into several subsystems or processes. One or several of these processes may be examined using piece models. A system may be represented as shown below:



Blocks A, B, and D may be represented by piece models and C may have assumed characteristics. It is easiest to assume steady state conditions for the block C not represented by a piece model.

As another example, a Markov Model could be used to analyze the branching points of a software program to determine the various paths of logic flow and the probability of the time and use of the various paths. The results could demonstrate a relationship to system workload.

LEVEL OF CONFIDENCE:

Confidence in any specific piece model will vary with the model, the data on which it is modeled, and the complexity of the system or subsystem being modeled.

CATEGORY OF TECHNIQUE: Analytical

TECHNIQUE: Boundaries

DESCRIPTION: This technique assumes that one can obtain reasonable sizing and timing estimates by evaluating systems or processes within certain boundaries, such as establishing upper and lower bounds on system performance requirements.

REQUIRED DATA: Identify the most utilized component, subsystem, or function in a system. Determine upper and lower bounds for its performance.

ASSUMPTIONS: It is assumed that the most critically used component, subsystem, or function sets the limit on overall system performance.

APPLICATION: First identify the most critically used component, subsystem, or function of a system. Based on other analytical methods or analogy, determine the maximum capability possible for that specific function. This would be the upper limit. Then determine the minimum capability that still meets the established goals for a system. This would be the lower limit. Compare these boundaries with the system being evaluated to determine if sizing and timing parameters are acceptable.

LEVEL OF CONFIDENCE: One's level of confidence depends on one's ability to identify the most utilized component, subsystem, or function in a system, and then to properly establish the upper and lower bounds for analysis.

CATEGORY OF TECHNIQUE: Analytical

TECHNIQUE: Queuing Theory

DESCRIPTION: Queuing theory is a mathematical study of waiting lines and encompasses concepts of service time, waiting time, and service disciplines. This technique is of value in evaluating critical points of a system such as data waiting in a buffer until other data in core has been processed.

REQUIRED DATA: Processes within a computer system are viewed as lines for service. An application of this technique requires a modeling of a system that breaks it into critical lines where the service discipline, service times, and arrival rates are either assumed or known.

ASSUMPTIONS: The service discipline, service times, and arrival rates must either be assumed or known for a system. Also, steady-state conditions must be assumed.

APPLICATION: The systems or subsystems examined must be reduced to lines with the characteristics specified. Many smaller computer configurations have already been modeled for analysis, and it is only necessary to assume the values of the parameters for each line. The following authors are but a few who discuss queuing theory and modeling in relation to computer systems: Ferrari 5/, Kobayashi 11/, Martin 13/, and Svobodova 19/.

LEVEL OF CONFIDENCE: Confidence in the results of the application of queuing theory will depend on one's experience and understanding of the technique.

CATEGORY OF TECHNIQUE: Models

TECHNIQUE: Deterministic

DESCRIPTION: This is in a class of simple models in which all variables have been determined rather than obtained by random selection.

REQUIRED DATA: Only the factors specified are required and no knowledge of distributions or non-steady-state situations is required.

ASSUMPTIONS: It is assumed that steady-state conditions prevail.

APPLICATION: Identify the elements of the system to be modeled, find the appropriate models, and supply the required parameters. The selection of a model depends on the type of results desired. For more information regarding deterministic models, refer to Ferrari 5/ and Kobayashi 11/.

LEVEL OF CONFIDENCE: Level of confidence will depend on the models selected and one's understanding of the models, particularly the constraints of the selected models.

CATEGORY OF TECHNIQUE: Models

TECHNIQUE: Probabilistic

DESCRIPTION: Probabilistic models can incorporate statistical fluctuations in arrival and service demand rates and times, in addition to the simple parameters used in deterministic models. The models are useful in demonstrating variations in workload.

REQUIRED DATA: The data required, in addition to simple system parameters, is a knowledge of the distribution of arrival and service demand rates and times of a system or subsystem.

ASSUMPTIONS: It is assumed that a model approximates the system being examined and that the nature of the required statistical distributions are known, or can be acquired.

APPLICATION: Identify the elements of the system to be modeled, find the appropriate models, and supply the required parameters. The selection of a model depends on objectives of the study, the knowledge of the required statistical distributions, and the results provided by selected models. Reference Ferrari 5/ for additional information.

LEVEL OF CONFIDENCE: With the proper application of this technique, one should obtain better results than those obtained from deterministic models.

CATEGORY OF TECHNIQUE: Simulations

TECHNIQUE: Deterministic

DESCRIPTION: A deterministic simulation is a numerical technique for modeling the behavior of a computer system without regard to probabilistic considerations. Deterministic simulations may be discrete, continuous, or workload driven. Discrete simulations are triggered by time. The use of Markov chains is an example of a discrete application. Continuous simulations are driven by events, but there is no specific time quantum. Workload simulations are dependent on workload conditions.

REQUIRED DATA: The system must be broken into subsystems that can be represented by a mathematical model. The model is then programmed. The results of the program are the characteristics of the system.

ASSUMPTIONS: It is assumed that the model is a good representation of the system and that steady-state conditions apply.

APPLICATION: The system is first modeled or fitted to a model already available. The parameters are varied to realize the variances in results. The model may be run several times with variations in parameters to yield a sensitivity analysis. The results are examined and are compared, depending on the assumed parameters. The results are evaluated and compared to other simulation results, if appropriate. For additional information regarding deterministic simulations, refer to Ferrari 5/.

LEVEL OF CONFIDENCE: The results should give a high degree of confidence, provided the parameters are properly applied.

CATEGORY OF TECHNIQUE:	Simulations
TECHNIQUE:	Stochastic
DESCRIPTION:	A stochastic or Monte Carlo simulation is a mathematical modeling of a computer system that includes characteristics of the system that are random in nature, such as a workload that can be described by probability distributions.
REQUIRED DATA:	The data required include basic parameters to drive the model and an understanding of the parameters that can be described by probability distributions.
ASSUMPTIONS:	It is assumed that the model and statistical distributions used are refined approximations of the system being evaluated.
APPLICATION:	The system is first modeled or fitted to a model already available. The parameters are varied to realize the variances in results. Also since the model generally will have random variables, a compilation of multiple runs is required for each set of variables. This will yield a set of statistics that better approximates the system. The results are examined and are compared depending on the assumed parameters and statistical distributions. See Ferrari <u>5</u> /, Kobayashi <u>11</u> /, and Svobodova <u>20</u> / for further discussions on this technique.
LEVEL OF CONFIDENCE:	Depending on the accuracy with which the parameters are established, one should have a medium to high degree of confidence in the results.

CATEGORY OF TECHNIQUE: Benchmarks

TECHNIQUE: Actual Workload

DESCRIPTION: Actual workload benchmarks are used to evaluate a system or subsystem using the actual programs of a computer system, under assumed conditions.

REQUIRED DATA: A specification of operating conditions, workload definition and the system being evaluated, are required before a benchmark can be conducted.

ASSUMPTIONS: It is assumed that the system or subsystem is in existence and will adequately handle the actual workload under the assumed conditions.

APPLICATION: Based on the actual workload requirements and using the actual programs, under assumed conditions, various performance and timing parameters can be determined. The technique does not require an in-depth understanding of specific job characteristics since one is using actual workload requirements. Reference Ferrari 5/, and Svobodova 20/ for more information.

LEVEL OF CONFIDENCE: Depending on approximation of the assumed conditions to the actual conditions, one should have a high level of confidence in the technique and its results.

CATEGORY OF TECHNIQUE: Benchmarks

TECHNIQUE: Synthetic

DESCRIPTION: A synthetic benchmark contains all the important ingredients of a system or subsystem except certain characteristics that are developed for the benchmarking effort, such as an estimated workload for selected conditions.

REQUIRED DATA: The parameters that might be addressed in a synthetic benchmark include such characteristics as CPU processing demands, storage requirements, disc and file access characteristics.

ASSUMPTIONS: It is assumed that only the critical characteristics need to be modeled to obtain reasonable results.

APPLICATION: Identify critical characteristics of the system or subsystem being evaluated. Develop estimated characteristics and combine into an overall system and execute. Ferrari 5/ and Svobodova 20/ discuss the use of this technique.

LEVEL OF CONFIDENCE: Provided the estimated characteristics used are reasonable, this technique can produce a medium to high level of confidence in its results.

CATEGORY OF TECHNIQUE: Monitors

TECHNIQUE: Hardware

DESCRIPTION: Hardware monitors may be either internal or external to a computer system and are able to detect and collect very refined data regarding performance characteristics of a system or subsystem.

REQUIRED DATA: Selective data, depending on the type of monitor used, is collected in a real-time mode.

ASSUMPTIONS: If actual or simulated programs and workloads are processed, hardware monitors will provide very refined data as compared to software monitors.

APPLICATION: Hardware monitors are either included in a system's hardware or may be interfaced subsequently. Since hardware monitors do not directly affect the data processing capabilities of a system, there is no impact on the results obtained as is the case with software monitors. Insure that one understands how to interpret results. For more information regarding this technique, see Ferrari 5/ and Svobodova 20/.

LEVEL OF CONFIDENCE: High level of confidence.

CATEGORY OF TECHNIQUE: Monitors

TECHNIQUE: Software

DESCRIPTION: Software monitors are special programs that are designed to collect data about a computer system's performance. Use of software monitors causes some distortion of performance data due to the execution time required by the software monitoring programs. Software monitors also impact on sizing parameters since they require memory space for the program and sometimes the data.

REQUIRED DATA: Selective data, depending on the software monitor's program, is collected as the program is executed.

ASSUMPTIONS: If actual or simulated programs and workloads are processed, software monitors will provide fairly refined data, but not as refined as hardware monitors. Also, there will be some distortion of results as mentioned above.

APPLICATION: Determine what sizing and timing parameters are to be measured. Acquire software monitors that will perform the measurements. Insure that one has a thorough understanding of how the monitoring results are to be interpreted. Perform the software monitoring and evaluate the results. In certain cases, some or all of the impacts of the software monitoring effort can be determined and the results refined accordingly. See Ferrari 5/ and Svobodova 20/.

LEVEL OF CONFIDENCE: High level of confidence, however, not as high as with hardware monitors.

CATEGORY OF TECHNIQUE: Monitors

TECHNIQUE: Hybrid

DESCRIPTION: Hybrid monitors are a combination of hardware and software monitors that are designed to collect performance data from a computer system. Certain parts of the data collection may be accomplished by either or both the hardware or software portions of a hybrid monitor.

REQUIRED DATA: Selective data, depending on the hybrid monitor used.

ASSUMPTIONS: If actual or simulated programs and workloads are processed, hybrid monitors will provide more refined data than will software monitors, though not as refined as hardware monitors. Also, there will be some distortion of results due to the impact of the software portion of the monitor.

APPLICATION: Determine what sizing and timing parameters are to be measured. Obtain hybrid monitors that will perform measurements. Insure that one has a thorough understanding of how the monitoring results are to be interpreted. Perform the monitoring and evaluate the results. For more information on hybrid monitors, see Ferrari 5/ and Svobodova 20/.

LEVEL OF CONFIDENCE: High level of confidence, though not as high as hardware monitors.

REFERENCES

1. Beizer, Boris, Micro-Analysis of Computer System Performance, Van Nostrand Reinhold Co., New York, 1978.
2. Capps, L. R., Software Management and the Testing of Weapons Systems that Contain an Embedded Computer System, Defense Systems Management College, Student Report 75-2, November 1975. AD/A-026-556
3. Doty, D. L., Nelson, P. J., Stewart, K. R., Software Cost Estimation Study - Guidelines for Improved Software Cost Estimating, Doty Associates, Inc., Report No. RADC-TR-77-220, Vol. II (with Errata), February 1977. AD/A-044-609
4. Embedded Computer Resources and the DSARC Process - A Guidebook, Office of the Secretary of Defense, 1977. AD/A-046-398
5. Ferrari, Domenica, Computer Systems Performance Evaluation, Prentice-Hall, Inc., Englewood Cliffs, N.Y., 1978.
6. Finfer, M. C., Bratman, H., Software Acquisition Management Guidebook: Verification, System Development Corporation, Report No. ESD-TR-77-263, August 1977. AD/A-048-577
7. Gilbert, Dennis M., Mulford, James O., et al., "Guidance for Sizing ADP Systems (ADPS's)," Computer Science & Technology: Computer Performance Evaluation Users Group (CPEUG), Proceedings of the Fourteenth Meeting, Boston, Mass., Oct. 24-27, 1978, James E. Weatherbee, ed., October 1978, pp. 305-330.
8. Halstead, Maurice H., Elements of Software Science, Elsevier North-Holland, Inc., New York, 1977.
9. Herd, J. H., Postak, J. N., Russell, W. E., et al., Software Cost Estimation Study - Study Results - Final Report, Doty Associates, Inc., Report No. RADC-TR-77-220, Vol. I (with Errata), February 1977. AD/A-042-264
10. Information Processing/Data Automation Implications of Air Force Command and Control Requirements in the 1980's (CCIP-85), Vol. IV - Technology Trends Software, SAMSO TR-72-122, October 1973. AD-919-267
11. Kobayashi, H., Modeling and Analysis: An Introduction to System Performance Evaluation Methodology, Addison-Wesley Publishing Co., Inc., Reading, Mass., 1978.

AD-A086 444

DOTY ASSOCIATES INC ROCKVILLE MD

HANDBOOK OF PROCEDURES FOR ESTIMATING COMPUTER SYSTEM SIZING AN--ETC(U)

FEB 80 W B HUMPHREY, J N POSTAK

F19628-79-C-0106

UNCLASSIFIED

DAI-TR-235-VOL-1

ESD-TR-80-115-VOL-1

ML

2 of 2

AD-A086-444

END

DATE

FILED

8-80

DTIC

12. Kossiakoff, A., et al., Dod Weapon Systems Software Management Study, Johns Hopkins University Applied Physics Laboratory, APL/JHU SR 75-3, June 1975.
13. Martin, James, Design of Real-Time Computer Systems, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1967.
14. Neil, George, Software Acquisition Management Guidebook: Reviews and Audits, System Development Corporation, Report No. ESD-TR-78-117, November 1977. AD/A-052-567
15. Operational Software Management and Development for U.S. Air Force Computer Systems, National Academy of Sciences, National Research Council, Air Force Studies Board, 1977.
16. "Papers Presented at the Conference on Simulation, Measurement and Modeling of Computer Systems," August 13-15, 1979, Boulder, Colorado, A Special Joint Issue; SIMULETTER, Vol. 11, No. 1, and Performance Evaluation Review, Vol. 8, No. 3, Fall 1979.
17. Roberts, Alan J., Command and Control System Acquisition: Software Implications User/Developer Interface, AIIE Conference on Electronic System Acquisition Management, Washington, D.C., 25-27 April 1979.
18. Skrukrud, Allan M., Willmorth, Norman E., Software Acquisition Management Guidebook: Validation and Certification, System Development Corporation, Report No. ESD-TR-77-326, August 1977. AD/A-053-039
19. Svobodova, L., Computer Performance Measurement and Evaluation Methods: Analysis and Applications, Stanford Electronics Laboratories, Report No. TR-72, June 1974. AD/A-013-318
20. Svobodova, L., Computer Performance Measurement and Evaluation Methods: Analysis and Applications, Vol. 2 in the Computer Design and Architecture Series, Elsevier Computer Science Library, American Elsevier Publishing Co., Inc., New York, 1976.

BIBLIOGRAPHY

1. Abridged Proceedings from the Software Management Conference, H. Al Richardson, Chairman, Software Conference Committee, American Institute of Aeronautics and Astronautics (A.I.A.A.), First Series, 22-23 March 1976.
2. Abridged Proceedings from the Software Management Conference, H. Al Richardson, Chairman, Software Conference Committee, American Institute of Aeronautics and Astronautics (A.I.A.A.), Series 1977, 27-28 January 1977.
3. Air Force ADP Experience Handbook (Pilot Version), Planning Research Corporation, ESD-TR-66-673, December 1966. AD-646-863
4. American National Dictionary for Information Processing, Federal Information Processing Standards Publication 11-1, National Bureau of Standards, September 1977.
5. Anderson, L. A., Software Management: Applying Hardware Disciplines to Software Management, Defense Systems Management College, November 1976. AD/A-041-887
6. Archibald, Russell D., Managing High-Technology Programs and Projects, John Wiley & Sons, Inc., New York, 1976.
7. Archibald, W. R., Manley, John H., Proceedings of the Aeronautical Systems Software Workshop, HQ AFSC/XRF, Andrews AFB, Report No. AFSC-TR-74-03, July 1974. AD/A-004-547
8. Baker, W. F., Software Data Collection and Analysis: A Real-Time System Project History, IBM Corporation, Report No. RADC-TR-77-192, June 1977. AD/A-041-644
9. Bamberger, F., Ostanek, W., Rye, P., Software Systems Development: A CSDL Project History, Charles Stark Draper Lab, Inc., Report No. RADC-TR-77-213, Vol. I, June 1977. AD/A-042-186
10. Barbacci, M. R., Burr, et al., Evaluation of Alternative Computer Architectures, Carnegie-Mellon University, Pittsburgh, Pa., Report No. AFOSR-TR-77-0381, February 1977. AD/A-038-462
11. Barry, Barbara S., Naughton, John J., Chief Programmer Team Operations Description - Structured Programming Series, IBM Corporation, Report No. RADC-TR-74-300, Vol. X, 22 January 1975.
12. Beizer, Boris, Micro-Analysis of Computer System Performance, Van Nostrand Reinhold Co., New York, 1978.

13. Bell, C. G., Mudge, J. C., McNamara, J. E., Computer Engineering - A DEC View of Hardware Systems Design, Digital Equipment Corporation, Bedford, Ma., 1978.
14. Bell, D. E., Sullivan, J. E., Further Investigations Into the Complexity of Software, Mitre Corporation, Report No. MTR-2874, Vol. II, June 1974.
15. Bell, T. E., Boehm, B. W., Watson, R. A., Computer Performance Analysis: Framework and Initial Phases for a Performance Improvement Effort, RAND Corporation, Report No. R-549-1-PR, November 1972.
16. Bell, T. E., Computer Performance Analysis: Measurement Objectives and Tools, RAND Corporation, Report No. R-584-NASA/PR, February 1971.
17. Bennett, Martha S., et al., Documentation Standards Addendum - Structured Programming Series, IBM Corporation, Report No. RADC-TR-74-300, Vol. VII (Addendum), 15 April 1975.
18. Benwell, Nicholas, Benchmarking - Computer Evaluation and Measurement, John Wiley & Sons, Inc., New York, 1975.
19. Bielski, J., Ewing, R., Transaction Processing Operating System (TPOS), Honeywell Information Systems, Inc., Report No. RADC-TR-77-276, August 1977. AD/A-044-612
20. Black, Rachael K. E., et al., BCS Software Production Data, Boeing Computer Services, Inc., RADC-TR-77-116, March 1977. AD/A-039-852
21. Bockian, James, An Overview of Project Management/Project Planning - Systems Development Management, 34-01-02, Auerbach Publishers, Inc., Pennsauken, N.J., 1976.
22. Boehm, B. W., Bosch, C. A., Liddle, A. S., Wolverton, R. W., Impact of New Technologies on Software Configuration Management, TRW Systems Group, June 1974.
23. Boehm, Barry, et al., Practical Strategies for Developing Large Software Systems, (Ellis Horowitz, ed.), Addison-Wesley Publishing Company, Inc., Reading, Mass., 1975.
24. Boehm, Barry W., et al., Characteristics of Software Quality, Vol. I in the TRW Series of Software Technology, North-Holland Publishing Company, New York, 1978.
25. Boehm, B. W., Computer Systems Analysis Methodology: Studies in Measuring, Evaluating, and Simulating Computer Systems, RAND Corporation, Report No. R-520-NASA, September 1970.

26. Boehm, B. W., Information Processing Requirements for Future Air Force Command and Control Systems and Some Implications for Software Research and Development, RAND Corporation, P-4795, March 1972.
27. Boehm, B. W., Software and Its Impact: A Quantitative Assessment - TRW Software Series, TRW, Report No. TRW-SS-73-04, May 1973.
28. Boehm, B. W., Some Steps Toward Formal and Automated Aids to Software Requirements Analysis and Design - TRW Software Series, TRW, Report No. TRW-SS-74-02, August 1974.
29. Bolen, N. E., An Air Force Guide to Contracting for Software Acquisition, Mitre Corporation, Report No. ESD-TR-75-365, January 1976. AD/A-020-444
30. Bourdon, Gerard A., Duquette, Joseph A., Computerized Model for Estimating Software Life-Cycle Costs, (Model Concept), ESD Hanscom AFB, Report No. ESD-TR-77-253, Vol. I, April 1978. AD/A-053-937
31. Boyse, J. W., Irani, K. B., Uppal, I. S., et al., Study of Information in Multiple-Computer and Multiple-Console Data Processing Systems, Michigan University, Annual Report #4, Report No. RADC-TR-071-160, August 1971. AD/A-729-194
32. Brooke, Barbara H., Survey of Mass Storage Systems, U.S. Army Engineer Topographic Laboratories, Report No. ETL-0082, 1 September 1975. AD/A-034-437
33. Brooks, Frederick P., Jr., The Mythical Man-Month: Essays on Software Engineering, Addison-Wesley Publishing Co., Inc., Reading, Mass., 1975.
34. Brown, J. R., Nelson, E. C., Functional Programming, TRW, Report No. RADC-TR-78-24, February 1978. AD/A-052-997
35. Bucciarelli, Marco A., Technical Performance Measurement for Computer Software Development Programs, Defense Systems Management College, Ft. Belvoir, Va., Study Report PMS 74-1, May 1974. AD/A-041-888
36. Bullen, Richard H., Software First Concepts - Engineering of Quality Software Systems, Mitre Corporation, Report No. RADC-TR-74-325, Vol. III, January 1975. AD/A-007-768
37. Capps, L. R., Software Management and the Testing of Weapons Systems that Contain an Embedded Computer System, Defense Systems Management College, Student Report 75-2, November 1975. AD/A-026-556

38. Carey, L. J., Hirschfield, G. A., Tucker, A. E., et al., Survey of Software Techniques - Spaceborne Software Systems Study, System Development Corporation, Report No. SSD-TR-67-11, Vol. IV, January 1967. AD-807-400
39. Carter, D. M., Gibson, H. L., Rademacher, R. A., A Study of Critical Factors in Management Information Systems for the U.S. Air Force, Colorado State University, Information Systems Series 460-2.
40. Carter, S., Donahoo, J., et al., Software Production Data, Computer Science Corporation-ISIS/RADC, Report No. RADC-TR-77-177, July 1977. AD/A-042-686
41. Caudill, Ray, Applying Program Management Concepts to Software Development: An AFR 300-15 Critique, Defense Systems Management College, Ft. Belvoir, Va., Study Report 77-2, November 1977. AD/A-052-126
42. Cheng, L., Sullivan, J. E., Case Studies in Software Design, Mitre Corporation, Report No. MTR-2874, Vol. I, June 1974.
43. Cheng, L. I., Some Case Studies in Structured Programming - Engineering of Quality Software Systems, Mitre Corporation, RADC-TR-74-325, Vol. VI, January 1975. AD/A-007-771
44. Chin, Yeh-Hao, et al., Improvement in a System's Throughput - From the Standpoint of File Organization and Searching Strategies, Texas University, AFOSR-TR-72-2014, September 1972. AD-757-495
45. Clapp, J. A., Software Error Classification Methodology - Engineering of Quality Software Systems, Mitre Corporation, Report No. RADC-TR-74-325, Vol. VII, January 1975. AD/A-007-772
46. Clapp, Judith A., A Review of Software Cost Estimation Methods, Mitre Corporation, Report No. MTR-3264, June 1976. AD/A-029-748
47. Clapp, Judith A., LaPadula, Leonard J., Engineering of Quality Software Systems, Mitre Corporation, Report No. RADC-TR-74-325, Vol. I, January 1975. AD/A-007-766
48. Computer Science and Technology: Computer Performance Evaluation User's Group (CPEUG Thirteenth Meeting), National Bureau of Standards, Report No. NBS-SP-500-18, September 1977. PB-272-072
49. Computer System Performance Measurement - Data Processing Manual, 5-03-03, Auerbach Publishers, Inc., Pennsauken, N.J., 1974.
50. Corrigan, Ann E., Results of an Experiment in the Application of Software Quality Principles, Mitre Corporation, Report No. MTR-2874, Vol. III, June 1974.

51. Current Trends in Programming Methodology, Vol. III - Software Modeling, Chandy, K. M., Yeh, R. T., eds., Prentice-Hall, Inc., Englewood Cliffs, N.J., 1978.
52. Daly, Edmund B., "Management of Software Development," IEEE Transactions on Software Engineering, May 1977, pp. 229-242.
53. Defense Management Journal, (Hardware/Software), Robert F. Goerlich, ed., Vol. XI, No. 4, Government Printing Office, October 1975.
54. DeLutis, Thomas G., A Methodology for the Performance Evaluation of Information Processing Systems, The Ohio State University Research Foundation, March 1977. PB-266-943
55. DeRoze, Barry C., DoD Defense System Software Management Plan, Department of Defense, March 1976.
56. Devenny, Thomas J., An Exploratory Study of Software Cost Estimating at the Electronic Systems Division, Air Force Institute of Technology (EN), Report No. GSM-SM-76S-4, July 1976. AD/A-030-162
57. DiMond, A. E., Understanding and Interpreting Hardware Reliability and Availability - Data Communications Management, 51-20-01, Auerbach Publishers, Inc., Pennsauken, N.J., 1976.
58. Doty, D. L., Nelson, P. J., Stewart, K. R., Software Cost Estimation Study - Guidelines for Improved Software Cost Estimating, Doty Associates, Inc., Report No. RADC-TR-77-220, Vol. II (with Errata), February 1977. AD/A-044-609
59. Drezner, Stephen M., Shulman, H., et al., Computer Resource Management Study: Executive Summary, RAND Corporation, Report No. R-1855-PR, September 1975. AD/A-018-884
60. Drezner, S. M., et al., The Computer Resource Management Study, RAND Corporation, Report No. R-1855/1-PR, April 1976. AD/A-042-204
61. Driscoll, Alan J., Software Visibility for the Program Manager, Defense Systems Management College, Project Report PMC 76-2, November 1976. AD-A035-164
62. Drossman, Melvyn M., Functional Software Development, New York Institute of Technology, Report No. AFOSR-TR-78-0110, November 1977. AD/A-050-294
63. Drummond, Mansford E., Jr., Evaluation and Measurement Techniques for Digital Computer Systems, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1973.

64. EDP Performance Management Handbook, Vol. 1: Audit and Control, Howard, Philip C., Stevens, Barry A., eds., Applied Computer Research, 1979.
65. EDP Performance Management Handbook, Vol. 2 - Tools and Techniques, Howard, Philip C., ed., Applied Computer Research, 1979.
66. Ely, E. H., Software Management: A Dynamic Approach, Defense Systems Management College, May 1977. AD/A-042-958
67. Embedded Computer Resources and the DSARC Process - A Guidebook, Office of the Secretary of Defense, 1977. AD/A-046-398
68. Encyclopedia of Computer Science, Ralston, Anthony, ed., Van Nostrand Reinhold Company, New York, 1976.
69. Engelman, C., Towards an Analysis of the LISP Programming Language - Engineering of Quality Software Systems, Mitre Corporation, Report No. RADC-TR-74-325, Vol. IV, January 1975. AD/A-007-769
70. England, G. R., Summers, W. P., New Directions in Avionic System Design, NAECON'77 National Aerospace & Electronics Conference, 17-19 May 1977.
71. Esposito, J. E., Statistical Analysis to Determine Digital Computer Workload Characteristics, Mitre Corporation, Report No. ESD-TR-74-175, June 1974. AD-786-061
72. Farnsworth, D. L., Hoffman, C. P., Shutt, J. J., Mass Memory Organization Study, Westinghouse Electric Company, Report No. RADC-TR-76-254, September 1976. AD/P-033-715
73. Farquhar, J. A., A Preliminary Inquiry Into the Software Estimation Process, RAND Corporation, Report No. RM-6271-PR, August 1970.
74. Farr, L., LaBolle, V., Willmorth, N., Planning Guide for Computer Program Development, System Development Corporation, Report No. TM-2314, May 1965.
75. Farr, L., Nanus, B., Factors that Affect the Cost of Computer Programming, System Development Corporation, Report No. TM-1447, July 1964. AD-603-707
76. Farrell, Dennis W., "Navy Airborne Weapon System Software Acquisition," Defense Systems Management Review, Vol. I, No. 6, Summer 1978, pp. 47-52.
77. Feldmeyer, Bruce A., Computer Performance Evaluation During System Acquisition, Mitre Corporation, Report No. ESD-TR-76-358, January 1977. AD/A-037-117

78. Ferrari, Domenica, Computer Systems Performance Evaluation, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1978.
79. Fife, Dennis W., Computer Science and Technology: Computer Software Management - A Primer for Project Management and Quality Control, National Bureau of Standards, Report No. NBS-SP-500-11, July 1977. PB-271-550
80. Finfer, M. C., Bratman, H., Software Acquisition Management Guidebook: Verification, System Development Corporation, Report No. ESD-TR-77-263, August 1977. AD/A-048-577
81. Finfer, M. C., Data Requirements for Productivity and Reliability Studies - Software Data Collection Study, System Development Corporation, Report No. RAD-TR-76-329, Vol. III, December 1976. AD/A-036-064
82. Finfer, M., Mish, R., Software Acquisition Management Guidebook: Cost Estimation and Measurement, System Development Corporation, Report No. ESD-TR-78-140, Vol. I, March 1978. AD/A-055-574
83. Finfer, M., Templeton, M., Compendium of Procedures and Parameters - Software Data Collection Study, System Development Corporation, Report No. RAD-TR-76-329, Vol. VII, December 1976. AD/A-036-247
84. Fisher, D. A., Common Programming Language for the Department of Defense-Background and Technical Requirements, Institute for Defense Analyses, Paper No. P-1191, June 1976. AD/A-028-297
85. Fleischer, R. J., Effects of Management Philosophy on Software Production - Engineering of Quality Software Systems, Mitre Corporation, Report No. RAD-TR-74-325, Vol. II, January 1975. AD/A-007-767
86. Fleishman, T., Current Results from the Analysis of Cost Data for Computer Programming, System Development Corporation, Report No. ESD-TR-66-462, August 1966. AD-637-801
87. Fleiss, Joel E., Phillips, Guy W., Statistics Gathering Package for the Jovial Language, Proprietary Software Systems, Inc., Report No. RAD-TR-73-381, January 1974. AD-775-380
88. Flynn, Michael, J., Studies in the Organization of Computer Systems, 1977 Progress Summary, Technical Note No. 121, Digital Systems Laboratory, Stanford University, November 1977.
89. Foster, Caxton C., Computer Architecture, 2nd ed., Computer Science Series, Van Nostrand Reinhold Company, New York, 1976.

90. Frederic, Brad C., A Provisional Model for Estimating Computer Program Development Costs, Tecolote Research, Inc., Report No. TM-7/Rev. 1, August 1974.
91. Freeman, Peter, Wasserman, A., A Tutorial - Software Design Techniques, IEEE Computer Society, Second Edition, 1977.
92. Fried, Louis, Estimating Time, Cost, and Resource Requirements: Systems Analysis - Systems Development Management, 34-01-05, Auerbach Publishers, Inc., Pennsauken, N.J., 1976.
93. Fryer, Richard E., "Computer Systems in the Navy," Defense Systems Management Review, Vol. 1, No. 6, Summer 1978, pp. 40-46.
94. Gifford, David K., Hardware Estimation of a Process-Primary Memory Requirements, Massachusetts Institute of Technology, Report No. ESD-TR-77-147, January 1977. AD/A-044-563
95. Gilbert, Dennis M., Mulford, James O., et al., "Guidance for Sizing ADP Systems (ADPS's)," Computer Science & Technology: Computer Performance Evaluation Users Group (CPEUG), Proceedings of the Fourteenth Meeting, Boston, Mass., Oct. 24-27, 1978, James E. Weatherbee, ed., October 1978, pp. 305-330.
96. Glore, J. B., Software Acquisition Management Guidebook: Life Cycle Events, ESD-TR-77-22, March 1977.
97. Glore, J. B., Bjerstedt, W. R., Software Acquisition Management Guidebook: Statement of Work Preparation, Mitre Corporation, Report No. ESD-TR-77-16, January 1977. AD/A-035-924
98. Glore, J. B., Friedman, M. P., Goheen, S. M., Software Acquisition Management Guidebook: Regulations, Specifications, and Standards, Mitre Corporation, Report No. ESD-TR-78-178, November 1978.
99. Goodman, Arnold F., Measurement of Computer Systems--An Introduction, McDonnell Douglas Astronautics Company, Paper WD-1905, July 1972. AD-787-104
100. Gorsuch, John R., A Specified Writing Technique - Systems Development Management, 35-04-06, Auerbach Publishers, Inc., Pennsauken, N.J., 1978.
101. Gossick, Major General Lee V., "Management of System Acquisition Programs in the Air Force," Defense Industry Bulletin, Summer 1971, pp. 23-29.

102. Gradwohl, Alan J., Beckwith, George S., et al., Phase II Final Report on Use of Air Force ADP Experience to Assist Air Force ADP Management: Vol. I - Summary, Conclusions, and Recommendations, ESD-TR-66-671, December 1966. AD-646-867
103. Gradwohl, Alan J., Wootan, Welford O., Jr., Phase II Final Report on Use of Air Force ADP Experience to Assist Air Force ADP Management: Vol. III - Concept and Plan, Planning Research Corporation, ESD-TR-66-671, December 1966. AD-641-868
104. Graver, C. A., et al., Cost Reporting Elements and Activity Cost Tradeoffs for Defense System Software: Vol. III - Executive Summary, Final Report, General Research Corporation, CR-1-721/1, May 1977.
105. Graver, C. A., et al., Cost Reporting Elements and Activity Cost Tradeoffs for Defense System Software - Study Results, General Research Corporation, Report No. ESD-TR-77-262, Vol. I, 11 May 1977. AD/A-053-020
106. Graver, C.A., et al., Cost Reporting Elements and Activity Cost Tradeoffs for Defense System Software, General Research Corporation, Report No. ESD-TR-77-262, Vol. II, 11 May 1977. AD/A-053-021
107. Grover, Perry H., Willmorth, Norman E., An Investigation of Programming Practices in Selected Air Force Projects, System Development Corporation, Report No. RADC-TR-77-182, June 1977. AD/A-041-678
108. Guidelines for Benchmarking ADP Systems in the Competitive Procurement Environment, Federal Information Processing Standards Publication 42-1, National Bureau of Standards, 15 May 1977.
109. Guidelines for the Measurement of Interactive Computer Service Response Time and Turnaround Time, Federal Information Processing Standards Publication 57, National Bureau of Standards, 1 August 1978.
110. Hagan, S. R., Knight, C. W., Air Force Guide for Monitoring and Reporting Software Development Status, Mitre Corporation, Report No. ESD-TR-75-85, Vol. I, September 1975. AD/A-016-488
111. Halstead, Maurice H., Elements of Software Science, Elsevier North-Holland, Inc., New York, 1977.
112. Herd, J. H., Postak, J. N., Russell, W. E., et al., Software Cost Estimation Study - Study Results - Final Report, Doty Associates, Inc., Report No. RADC-TR-77-220, Vol. I (with Errata), February 1977. AD/A-042-264

113. Holub, Donald, Collecting Performance Data - Data Center Operations Management, 45-01-03, Auerbach Publishers, Inc., Pennsauken, N.J., 1977.
114. Hunter, Beverly, Conducting System Evaluation Reviews - Systems Development Management, 37-01-01, Auerbach Publishers, Inc., Pennsauken, N.J., 1977.
115. IEEE Transactions on Software Engineering, IEEE Computer Society, James H. Carter, ed., IEEE Vol. 2, 1976.
116. Index - TRW Software Series, F. C. Manthey, ed., TRW Systems Group, TRW-SS-INDEX, Issue No. 3, January 1976.
117. Information Processing/Data Automation Implications of Air Force Command and Control Requirements in the 1980's (CCIP-85), Vol. IV - Technology Trends Software, SAMSO TR-72-122, October 1973. AD-919-267
118. Information Processing/Data Automation Implications of Air Force Command and Control Requirements in the 1980's (CCIP-85), Vol. VII - Technology Trends: Integrated Design, SAMSO TR-72-122, January 1973. AD-906-757
119. Information Processing/Data Automation Implications of Air Force Command and Control Requirements in the 1980's, (CCIP-85), Volume X, Current R and D, Revised, SAMSO, October 1973. AD-768-979
120. Information Processing/Data Automation Implications of Air Force Command and Control Requirements in the 1980's (CCIP-85), Vol. XI, Roadmaps, SAMSO/XRS-71-1, May 1972. AD-902-515
121. The Information Processing Series, Vol. I - Database Management Systems, (from the National Computer Conferences), AFIPS Press, Shneiderman, Ben, ed., Montvale, N.J., 1976.
122. Irani, K. B., Hinshaw, D. L., et al., Study of Information in Multiple-Computer and Multiple-Console Data Processing Systems, Michigan University, Final Report 4/69-6/72, Report No. RADC-TR-72-250, October 1972. AD-751-276
123. Irani, K. B., Bauer, M. F., et al., Study of Information in Multiple-Computer and Multiple-Console Data Processing Systems, RADC Project Engineering, Report No. RADC-TR-75-276, Vol. I, November 1975. AD/A-019-202
124. James, Thomas G., Software Cost Estimating Methodology, System Evaluation Group, Report No. AFAL-TR-77-66, August 1977. AD/A-048-192

125. Jelinski, Z., Moranda, P. B., Metrics of Software Quality, Mac-Donnell Douglas Astronautics Company, Phase I, Report No. MDC-G7517, July 1978.
126. Jensen, Randall W., Tonies, Charles C., et al., Ready-to-Apply Techniques for Cost Effective Management, Design, Implementation, and Verification in All Phases of Software Engineering, Prentice-Hall, Inc., Englewood Cliffs, N.J., January 1979.
127. Jensen, Randall W., Tonies, Charles C., Software Engineering, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1979.
128. Johnson, L. A., et al., "Automated Analysis of System Specifications," Second U.S. Army Software Symposium (Proceedings), U.S. Army Computer Systems Command, Williamsburg, Va., 25-27 October 1978, pp. 235-258.
129. Jucevic, E., Airborne Systems Software Acquisition Engineering Guidebook for Regulations, Specifications, and Standards, Final Report, TRW-30323-6004-TU-00, November 1977. AD/A-058-428
130. Kessler, Marvin M., Kister, William E., Software Tool Impact-Final Report - Structured Programming Series, IBM Corporation, Report No. RADC-TR-74-300, Vol. XIV, 22 May 1975.
131. Kessler, Marvin M., Tinanoff, et al., Programming Language Standards - Structured Programming Series, IBM Corporation, Report No. RADC-TR-74-300, Vol. 1, 15 March 1975.
132. Kessler, Marvin M., Training Materials - Structured Programming Series, IBM Corporation, Report No. RADC-TR-74-300, Vol. XII, 15 July 1975.
133. Kimbleton, Stephen R., An Analytic Framework for Computer System Sizing and Tuning, RAND Corporation, P-5162, January 1974.
134. Knight, B. M., "Software Quality and Productivity," Defense Systems Management Review, Vol. 1 Summer 1978, pp. 54-65.
135. Kobayashi, H., Modeling and Analysis: An Introduction to System Performance Evaluation Methodology, Addison-Wesley Publishing Co., Inc., Reading, Mass., 1978.
136. Kokko, T. P., Modern Programming Practices In Command and Control Maintenance, (Student Research Report), Air Command and Staff College, Report No. 1310-78, April 1978. AD/B-030-624
137. Korthage, Robert R., The Information Technology Series, Vol. IV - Computer Networks and Communication, (from the National Computer Conferences), AFIPS Press, Montvale, N.J., 1978.

138. Kossiakoff, A., et al., DoD Weapon Systems Software Management Study, Johns Hopkins University Applied Physics Laboratory, APL/JHU SR 75-3, June 1975.
139. Kosy, Donald W., Air Force Command and Control Information Processing in the 1980's: Trends in Software Technology, The RAND Corporation, Report No. R-1012-PR, Vol. I, June 1974. AD/A-017-128
140. Kuck, David J., Computer System Capacity Fundamentals, National Bureau of Standards, Report No. COM-74-51065, October 1974.
141. Kuck, David J., The Structure of Computers and Computations, Vol. 1, John Wiley & Sons, Inc., New York, 1978.
142. Laemmel, A., Shooman, M., Statistical (Natural) Language Theory and Computer Program Complexity - Software Modeling Studies, Polytechnic Institute of New York, Report No. RADC-TR-78-4, Vol. II, April 1978. AD/A-054-900
143. LaPadula, L. J., Software Reliability Modeling and Measurement Techniques - Engineering of Quality Software Systems, Mitre Corporation, Report No. RADC-TR-74-325, Vol. VIII, January 1975. AD/A-007-773
144. Large Scale Information Systems, Syracuse University, Report No. RADC-TR-78-43, Vol. II, March 1978. AD/A-054-943
145. Large Scale Information Systems, Syracuse University, Report No. RADC-TR-78-43, Vol. III, March 1978. AD/A-054-944
146. Lehman, John H., Thayer, Richard H., Software Engineering Project Management: A Survey Concerning U.S. Aerospace Industry Management of Software Development Projects (Interim Report), Sacramento Air Logistics Center/ACD, Report No. ACD TR-77-02, November 1977. AD/A-050-802
147. Lipow, M., Application of Algebraic Methods to Computer Program Analysis - TRW Software Series, TRW, Report No. TRW-SS-73-10, May 1973.
148. Lipow, M., Manley, John H., Findings and Recommendations of the Joint Logistics Commanders Software Reliability Work Group (SRWG Report) (Executive Summary), HQ AFSC/XRF, Andrews AFB, Report No. HQ-AFSC-TR-75-05, Vol. I, November 1975. AD/A-018-881
149. Luppino, Fred M., Smith, R. L., Programming Support Library (PSL) Functional Requirements - Structured Programming Series, IBM Corporation, Report No. RADC-TR-74-300, Vol. V, 25 July 1974. AD/A-003-339

150. Luppino, Fred M., Tinanoff, N., Programming Support Library (PSL) Program Specifications - Structured Programming Series, IBM Corporation, Report No. RADC-TR-74-300, Vol. VI, 22 November 1974.
151. Madnick, Stuart E., Storage Hierarchy Systems, Massachusetts Institute of Technology, Project MAC - Interim Scientific Report, MAC-TR-107, April 1973. AD-760-001
152. Mamrak, Sandra A., DeRuyter, Patricia A., "Statistical Methods for Comparing Computer Services," Computer, November 1977, pp. 32-39.
153. Marciniak, John J., "Software Acquisition Within Air Force Systems Command - A Management Approach," Defense Systems Management Review, Vol. I, No. 6, Summer 1978, pp. 32-39.
154. Maron, M. E., Computers and Our Future, RAND Corporation, P-3501, December 1966.
155. Martin, James, Design of Real-Time Computer Systems, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1967.
156. Mathis, N. S., Willmorth, W. E., Software Milestone Measurement Study, Naval Electronics Laboratory Center, Report No. NELC-TD-285, 7 November 1973. AD-775-305
157. Matick, Richard, Computer Storage Systems & Technology, John Wiley & Sons, Inc., New York, 1977.
158. Maule, Charles R., Long, M. E., "Performance Monitor and Data Acquisition System for Real-Time Embedded Computers," Proceedings of the IEEE 1977 National Aerospace and Electronics Conference, Dayton, Ohio, 17-19 May 1979.
159. McCall, Jim A., Richards, Paul K., et al., Concept and Definitions of Software Quality - Factors in Software Quality, General Electric Company, Report No. RADC-TR-77-369, Vol. I, November 1977. AD/A-049-014
160. McCall, Jim A., Richards, Paul K., et al., Metric Data Collection and Validation - Factors in Software Quality, General Electric Company, Report No. RADC-TR-77-369, Vol. II, November 1977. AD/A-049-015
161. McCall, Jim A., Richards, Paul K., et al., Preliminary Handbook on Software Quality for an Acquisition Manager - Factors in Software Quality, General Electric Company, Report No. RADC-TR-77-369, Vol. III, November 1977. AD/A-049-055
162. McGlynn, Daniel R., Distributed Processing and Data Communications, John Wiley & Sons, Inc., New York, 1978.

163. Meadow, Charles T., Applied Data Management, Information Science Series, John Wiley & Sons, Inc., New York, 1976.
164. Metzger, Philip, W., FSD Programming Project Management Guide, IBM, Vol. I, Report No. FSC-70-0243, April 1970.
165. Miller, Stephen W., ed., The Information Technology Series, Vol. II - Memory and Storage Technology, (from the National Computer Conferences), AFIPS Press, Montvale, N.J., 1977.
166. Miller E. F., Methodology for Comprehensive Software Testing, General Research Corporation, Report No. RADC-TR-75-161, June 1975. AD/A-013-111
167. Mish, Russell K., Software Acquisition Management Guidebook: Series Overview, System Development Corporation, Report No. ESD-TR-78-141, March 1978.
168. Morris, J. M., Klingbail, K. N., Kane, P., et al., Structured Programming Evaluation, Pattern Analysis and Recognition Corporation, Report No. RADC-TR-75-179, July 1975. AD/A-015-763
169. Myers, Ware, "Statistical Approach to Scheduling Software Development," Computer, Vol. XI, No. 12, December 1978, pp. 23-25.
170. Navarro, Aaron B., Romanczuk, Ronald J., Imbedded Software Monitors and Data Collection - Design and Implementation, PRC Information Sciences Company, Report No. RADC-TR-74-217, Vol. I, September 1974. AD-787-672
171. Navarro, Aaron B., Romanczuk, Ronald J., Imbedded Software Monitors and Data Collection - System and Functional Program Description, PRC Information Sciences Company, Report No. RADC-TR-74-217, Vol. II, September 1974. AD-787-673
172. Neil, George, Software Acquisition Management Guidebook: Reviews and Audits, System Development Corporation, Report No. ESD-TR-78-117, November 1977. AD/A-052-567
173. Neil, G., Gold, H., I., Software Acquisition Management Guidebook: Software Quality Assurance, System Development Corporation, Report No. ESD-TR-787-255, August 1977. AD/A-047-318
174. Nelson, Gary J., "Putting Computer Software to the Test," Defense Management Journal, Vol. XV, No. 2, March-April 1979, pp. 38-41.
175. Operational Software Management and Development for U.S. Air Force Computer Systems, National Academy of Sciences, National Research Council, Air Force Studies Board, Washington, D.C., 1977.

176. Ortega, Louis H., Documentation Standards - Structured Programming Series, IBM Corporation, Report No. RADC-TR-74-300, Vol. VII, 21 September 1974.
177. Overview of Software Development and Management - Management Guide to Avionics Software Acquisition, Logicon Technical Staff, Logicon Incorporated, Report No. ASD-TR-76-11, Vol. I, June 1976. AD/A-030-591
178. "Papers Presented at the Conference on Simulation, Measurement and Modeling of Computer Systems," August 13-15, 1979, Boulder, Colorado, A Special Joint Issue; SIMULETTER, Vol. 11, No. 1, and Performance Evaluation Review, Vol. 8, No. 3, Fall 1979.
179. Pariseau, Richard J., Improved Software Productivity for Military Computer Systems Through Structured Programming, Naval Air Development Center, Report No. NADC-76044-50, March 1976. AD-A022-284
180. Patterson, Alton E., Embedded Computer System Software Management and Support After System Deployment, Defense Systems Management College, Report No. PMC 77-1, May 1977. AD/A-042-776
181. Performance of Computer Installations: Evaluation and Management, Ferrari, Domenica, ed., (Proceedings of the International Conference on the Performance of Computer Installations, ICPCI 78, June 22-23, 1978, Gardone Riviera, Lake Garda, Italy), North-Holland Publishing Co., New York, 1978.
182. Perry, Grover H., Willmorth, Norman E., Investigation of Programming Practices in Selected Air Force Projects, System Development Corporation, Report No. RADC-TR-77-182, June 1977. AD/A-041-678
183. Peterson, D. R., Software Acquisition Management Guidebook: Software Development and Maintenance Facilities, Mitre Corporation, Report No. ESD-TR-77-130, April 1977. AD/A-038-234
184. Piligian, M. S., et al., Configuration Management of Computer Program Contract End Items, Electronic Systems Division, L. G. Hanscom Field, Mass., ESD-TR-68-107, January 1968. AD-666-652
185. Pokorney, Joseph L., Mitchell, Wallace E., A Systems Approach to Computer Programs, ESD-TR-67-205, February 1967.
186. Pontius, Harry E., Life-Cycle Guidelines for Weapon System Software Management, Defense Systems Management College, Report No. PMC 76-1, 1976. AD/A-029-376
187. Prentiss, Nelson H., Viking Software Data, Martin Marietta Corporation, Report No. RADC-TR-77-168, May 1977. AD/A-040-770

188. Putnam, Lawrence H., "General Empirical Solution to the Macro Software Sizing and Estimating Problem," IEEE Transactions on Software Engineering, IEEE, Report No. 0098-5589, Vol. SE-4, July 1978, pp. 345-361.
189. Putnam, Lawrence H., "Progress in Modeling the Software Life-Cycle in a Phenomenological Way to Obtain Engineering Quality Estimates and Dynamic Control of the Process," Second Software Life-Cycle Management Workshop, U.S. Army Computer Systems Command, 21-22 August 1978 (Atlanta, Georgia), pp. 105-127.
190. Putnam, Lawrence H., Wolverton, Ray W., Quantitative Management: Software Cost Estimating, Computer Software and Applications Conference-Chicago, Institute of Electrical and Electronics Engineers, 8-11 November 1977.
191. Quantitative Software Models - Software Engineering Research Review, IIT Research Institute, DACS, RADC, SRR-1, March 1979.
192. Randall, L. Scott, A Relational Model of Data for the Determination of Optimum Storage Structures, University of Michigan, RADC-TR-72-25, February 1972. AD-740-581
193. Rao, Jai R., Memory-Use Estimator Function of a Program Executing in Paging Environment, California University, Report No. AFOSR-TR-74-0010, November 1973. AD-772-415
194. Roberts, Alan J., Command and Control System Acquisition: Software Implications User/Developer Interface, AIIE Conference on Electronic System Acquisition Management, Washington, D.C., 25-27 April 1979.
195. Rogers, John G., Writing Structured Programs for Virtual Systems - Computer Programming Management, 14-03-03, Auerbach Publishers, Inc., Pennsauken, N.J., 1976.
196. Royce, W. W., Managing the Development of Large Software Systems: Concepts and Techniques - TRW Software Series, TRW, Report No. TRW-SS-70-01, August 1970.
197. Ruston, H., Shooman, M., L., Summary of Technical Progress - Software Modeling Studies, Polytechnic Institute of New York, Report No. RADC-TR-75-245, September 1975. AD/A-018-618
198. Ruston, H., Shooman, M., L., Summary of Technical Progress - Software Modeling Studies, Polytechnic Institute of New York, Report No. RADC-TR-76-143, May 1976.
199. Ruston, H., Shooman, M., L., Summary of Technical Progress - Software Modeling Studies, Polytechnic Institute of New York, Report No. RADC-TR-77-88, March 1977. AD/A-038-508

200. Ruston, H., Shooman, M., L., Summary of Technical Progress - Software Modeling Studies, Polytechnic Institute of New York, Report No. RADC-TR-78-4, January 1978. AD/A-052-615
201. Schick, G. J., Wolverton, R. W., Assessment of Software Reliability - TRW Software Series, TRW, Report No. TRW-SS-72-04, September 1972.
202. Schoeffel, W. L., An Air Force Guide to Software Documentation Requirements, Mitre Corporation, Report No. ESD-TR-76-159, June 1976. AD/A-027-051
203. Schutzer, Daniel, "Command, Control, and Communications - Some Design Aspects," Information Systems - Effectiveness for the User - Proceedings, ACM and NBS Eighteenth Annual Technical Symposium, 21 June 1979, pp. 197-202.
204. Searle, Lloyd V., An Air Force Guide to Computer Program Configuration Management, System Development Corporation, Report No. ESD-TR-77-254, August 1977. AD/A-047-308
205. Searle, Lloyd V., An Air Force Guide to Computer Program Development Specification, System Development Corporation, Report No. ESD-TR-78-139, November 1977. AD/A-055-573
206. Second Software Life Cycle Management Workshop, Victor R. Basili, Chairman, U.S. Army Institute for Research in Management Information and Computer Science, Atlanta, Georgia, August 21-22, 1978.
207. Second U.S. Army Software Symposium, (Proceedings), U.S. Army Computer Systems Command, Williamsburg, Va., 25-27 October 1978.
208. Shen, John T., Analysis of Hardware and Software Storage and Retrieval Functions, Naval Electronics Laboratory Center, Report No. NELC-TD-259, 10 July 1973. AD-912-632
209. Shooman, M. L., Natarjan, S., Effect of Manpower Deployment and Bug Generation on Software Error Models, Polytechnic Institute of New York, Report No. RADC-TR-76-400, January 1977. AD/A-036-106
210. Sippl, Charles J., Sippl, Charles P., Computer Dictionary and Handbook, Howard W. Sams and Company, Inc., Indianapolis, Indiana, 1972.
211. Skrukrud, Allan M., Stanfield, J. R., Software Acquisition Management Guidebook: Software Maintenance, System Development Corporation, Report No. ESD-TR-77-327, October 1977. AD/A-053-040
212. Skrukrud, Allan M., Willmorth, Norman E., Software Acquisition Management Guidebook: Validation and Certification, System Development Corporation, Report No. ESD-TR-77-326, August 1977. AD/A-053-039

213. Smith, Ronald L., Estimating Software Project Resource Requirements - Structured Programming Series, IBM Corporation, Report No. RADC-TR-74-300, Vol. XI, January 1975.
214. Smith, Ronald L., Management Data Collection and Reporting-Final Report - Structured Programming Series, IBM Corporation, Report No. RADC-TR-74-300, Vol. IX, 24 October 1974.
215. Smith, Ronald L., Naughton, John J., Final Report - Structured Programming Series, IBM Corporation, Report No. RADC-TR-74-300, Vol. XIII, 15 July 1975.
216. Smith, Ronald L., Tinanoff, Nathan, Program Design Study - Structured Programming Series, IBM Corporation, Report No. RADC-TR-74-300, Vol. VIII, 30 May 1975.
217. Smith, Ronald L., Validation and Verification Study - Structured Programming Series, IBM Corporation, Report No. RADC-TR-74-300, Vol. XV, 22 May 1975.
218. Software Acquisition Process - Management Guide to Avionics Software Acquisition, Logicon Technical Staff, Logicon Incorporated, Report No. ASD-TR-76-11, Vol. II, June 1976. AD/A-030-592
219. Software Engineering: Its Development and Standards, Science Applications, Inc., Vol. I, January 1978. AD/A-049-869
220. Software Phenomenology - Working Papers of the Software Life-Cycle Management Workshop-Arlie House, U.S. Army Institute for Research in Managment Information and Computer Science, 21-23 August 1977.
221. Stover, Robert E., Statistics Collection Package for the Jovial J3 Programming Language, Rome Air Development Center (ISIS), Report No. RADC-TR-77-293, September 1977. AD/A-044-947
222. Sukert, Alan N., A Software Reliability Modeling Study, Rome Air Development Center (ISIS), RADC-TR-76-247, August 1976.
223. Sullivan, J. E., Engineering of Quality Software Systems - Measuring the Complexity of Computer Software, Mitre Corporation, Report No. RADC-TR-74-325, Volume V, January 1975. AD/A-007-770
224. Summary Notes of a Government/Industry Software Sizing and Costing Workshop, (In-House), Electronic Systems Division (AFSC), Report No. ESD-TR-76-166, 1-2 October 1974. AD-A026-964
225. Summary of Software Related Standards and Regulations - Management Guide to Avionics Software Acquisition, Logicon Technical Staff, Logicon Incorporated, ASD-TR-76-11, Vol. III, June 1976. AD/A-030-593

226. Svobodova, L., Computer Performance Measurement and Evaluation Methods: Analysis and Applications, Stanford Electronics Laboratories, Report No. TR-72, June 1974. AD/A-013-318
227. Svobodova, L., Computer Performance Measurement and Evaluation Methods: Analysis and Applications, Vol. 2 in the Computer Design and Architecture Series, Elsevier Computer Science Library, American Elsevier Publishing Co., Inc., New York, 1976.
228. Taylor, Harold W., Software Management: Current Trends and a Navy Application, Defense Systems Management College, Report No. PMC-77-1, May 1977. AD/A-042-937
229. Technical Aspects Relative to Software Acquisition - Management Guide to Avionics Software Acquisition, Logicon Technical Staff, Logicon Incorporated, Report No. ASD-TR-76-11, Vol. IV, June 1976. AD/A-030-594
230. Tinanoff, Nathan, Precompiler Specifications - Structured Programming Series, IBM Corporation, RADC-TR-74-300, Vol. II, 9 May 1979.
231. Transactions on Software Engineering - IEEE, IEEE Computer Society, Raymond T. Yeh, Editor; The Institute of Electrical and Electronics Engineers, Vol. SE-3, No. 1, January 1977.
232. Trimble, John T., Data Structuring Study-Final Report - Structured Programming Series, IBM Corporation, Report No. RADC-TR-74-300, Vol. IV, 21 April 1975.
233. Turn, Rein, Computer Systems Technology Forecast, RAND Corporation, P-5344, January 1975.
234. Turn, Rein, Computers in the 1980s, Columbia University Press, New York, 1974.
235. Turn, Rein, Computers In the 1980s - Trends In Hardware Technology, RAND Corporation, P-5189, March 1974. AD-783-323
236. Turn, R., Sine, B., Air Force Command and Control Information Processing in the 1980's: Trends in Hardware Technology, The RAND Corporation, Report No. R-1001-PR, Vol. V, October 1972. AD-907-626
237. Tzudiker, Harvey, "Software Configuration Management Testability and Traceability," Defense Systems Management Review, Vol. 1, Summer 1978, pp. 24-31.
238. Walston, C. E., Felix, C. P., "A Method of Programming Measurement and Estimation," IBM Systems Journal, Vol. 16, No. 1, 1977, pp. 54-73.

239. Walter, Larry R., Computer Resources Acquisition and Support for Air Force Weapons System, Defense Systems Management School, Report No. DSMS-PMC-750-1, Vol. I, May 1975. AD/A-027-675
240. Ware, W. H., Future Computer Technology and Its Impact, RAND Corporation, P-3279, March 1966.
241. Ware, W. H., Limits In Computing Power, RAND Corporation, P-4710, October 1971.
242. Weik, Martin H., Standard Dictionary of Computers & Information Processing, rev. 2nd ed., Hayden Book Co., Inc., Rochelle Park, N.J., 1977.
243. Welch, Terry A., "Analysis of Memory Hierarchies for Sequential Data Access," Computer, Vol. 12, No. 5, May 1979, pp. 19-26.
244. Willman, H. E., Jr., Beauregard, A. A., et al., Software Systems Reliability: A Raytheon Project History, Raytheon Company, Report No. RADC-TR-77-188, June 1977. AD/A-040-992
245. Willmorth, N.E., Finfer, M. C., Templeton, M. P., Software Data Collection Study - Summary and Conclusions, System Development Corporation, Report No. RADC-TR-76-329, Vol. I, December 1976. AD/A-036-115
246. Willmorth, N.E., Finfer, M. C., Templeton, M. P., Software Data Collection Study: An Analysis of Software Data Collections Problems and Current Capabilities, System Development Corporation, Report No. RADC-TR-76-329, Vol. II, December 1976. AD/A-036-116
247. Willmorth, N. E., Software Data Collection Study - Survey of Project Managers, System Development Corporation, Report No. RADC-TR-76-329, Vol. V, December 1976. AD/A-306-066
248. Willmorth, N. E., Software Data Collection Study Glossary, System Development Corporation, Report No. RADC-TR-76-329, Vol. VIII, December 1976. AD/A-036-265
249. Willoughby, W. J., "Software Reliability by Design: A Critical Need," Defense Systems Management Review, Vol. 1, Summer 1978, pp. 59-68.
250. Wolverton, R. W., Cost of Developing Large Scale Software - TRW Software Series, TRW, Report No. TRW-SS-72-01, March 1972.
251. Wolverton, R. W., Paradoxes in Management: Software Standards and Procedures - TRW Software Series, TRW, Report No. TRW-SS-74-11, April 1974.

252. Woolf, Ashby M., Analysis and Optimization of Multiprogrammed Computer Systems Using Storage Hierarchies, University of Michigan, RADC-TR-71-165, August 1971. AD-730-325
253. Yuhasz, R. S., Utterback, H. K., Newcomer, M. R., System of Software for the Triad Flight Computer, Johns Hopkins University, Report No. TG-1217, May 1973. AD-766-455

ACRONYMS AND ABBREVIATIONS

ADA	Proposed DoD High Level Language (not an acronym)
ADL	Authorized Data List
ADP	Automatic Data Processing
ADPE	Automatic Data Processing Equipment
ADPS	Automated Data Processing System
AFLC	Air Force Logistics Command
AFPRO	Air Force Plant Representation Office
AFR	Air Force Regulation
AFSC	Air Force Systems Command
AFSCM	Air Force Systems Command Manual
AFSCR	Air Force Systems Command Regulation
AFTEC	Air Force Test and Evaluation Center
ALGOL	ALGOrithmic Oriented Language
AMSDL	DoD Acquisition Management Systems and Data Requirements Control List
AMSL	Acquisition Management Systems List
ANSI	American National Standards Institute
APP	Advance Procurement Plan
APT	Automatically Programmed Tool
ATLAS	Abbreviated Test Language for All Systems
ASCII	American Standard Code for Information Interchange
ASD	Aeronautical Systems Division
ASPR	Armed Services Procurement Regulation
ATC	Air Training Command
ATE	Automatic Test Equipment
BASIC	Beginner's All Purpose Symbolic Instruction Code
BCD	Binary Coded Decimal
BIT	BiNary digiT
C2	Command and Control
C3	Command, Control, and Communications
C3I	Command, Control, Communications, and Intelligence
CCB	Configuration Control Board
CCBD	Configuration Control Board Directive
CDR	Critical Design Review
CDRL	Contract Data Requirements List
CFE	Contractor Furnished Equipment
CFI	Contractor Furnished Information
CFM	Contractor Furnished Materiel
CI	Configuration Item
CM	Configuration Management
COBOL	COmmon Business Oriented Language
CPC	Computer Program Component
CPCI	Computer Program Configuration Item

CPCSB	Computer Program Configuration Sub-Board
CPDP	Computer Program Development Plan
CPIN	Computer Program Identification Number
CPM	Critical Path Method
CPS	Characters Per Second
CPU	Central Processing Unit
CRISP	Computer Resources Integrated Support Plan
CRT	Cathode Ray Tube (Display)
CRWG	Computer Resources Working Group
CWBS	Contract Work Breakdown Structure
C/SCSC	Cost/Schedule Control Systems Criteria
DAR	Data Automation Requirement
DAR	Defense Acquisition Regulations
DBMS	Data Base Management System
DCASR	Defense Contract Administration Services Region
DCP	Decision Coordination Paper
DI	Data Item
DID	Data Item Description
DoD	Department of Defense
DoDD	Department of Defense Directive
DoDI	Department of Defense Instruction
DMO	Data Management Office
DSARC	Defense Systems Acquisition Review Council
DTC	Design to Cost
DT&E	Development, Test, and Evaluation
EAM	Electrical Accounting Machine
ECP	Engineering Change Proposal
ECS	Embedded Computer System(s)
EDP	Electronic Data Processing
ESD	Electronic Systems Division
ESDM	Electronic Systems Division Manual
FCA	Functional Configuration Audit
FEDSIM	Federal Computer Performance Evaluation and Simulation Center
FORTTRAN	FORmula TRANslator (Language)
FOT&E	Follow-On Operational Test and Evaluation
FQR	Formal Qualification Review
FQT	Formal Qualification Test
FSD	Full Scale Development
GFE	Government Furnished Equipment
GFI	Government Furnished Information
GFM	Government Furnished Materiel

HIPO	Hierarchical Input-Process-Output
HLL	High Level Language
HOL	High Order Language
H/W	Hardware
ICD	Interface Control Drawing
ICS	Interpretive Computer Simulation
ICWG	Interface Control Working Group
IDP	Integrated Data Processing
I/O	Input/Output
IOC	Initial Operational Capability
IOC	Input/Output Controller
IOT&E	Initial Operational Test and Evaluation
IPL	Initial Program Loader
IV&V	Independent Verification & Validation
JCL	Job Control Language
JOVIAL	Jules' Own Version of the International Algebraic Language
K	Thousand
LASER	Light Amplification by Stimulated Emission of Radiation
LCC	Life-Cycle Costs
LPM	Lines Per Minute
LSI	Large Scale Integrated (Circuit)
M	Million
MIPS	Million Instructions Per Second
MIS	Management Information System
MOL	Machine-Oriented Language
MTBF	Mean-Time-Between-Failures
MTTR	Mean-Time-to-Repair
NORAD	NORTH American Air Defense command
OCR	Optical Character Recognition
O/S	Operating System
O/S CMP	Operational/Support Configuration Management Procedures
OSD	Office of Secretary of Defense
OT&E	Operational Test and Evaluation

PCA	Physical Configuration Audit
PCM	Punched Card Machine
PCO	Procuring Contracting Officer
PDR	Preliminary Design Review
PERT	Program Evaluation and Review Technique
PL/I	Programming Language I
PM	Program Manager
PM	Program Memoranda
PMD	Program Management Directive
PMP	Program Management Plan
PMRT	Program Management Responsibility Transfer
PO	Program Office
POL	Procedure-Oriented Language
POM	Program Objective Memorandum
PQT	Preliminary Qualification Test
QA	Quality Assurance
QAM	Quality Assurance Manager
QAR	Quality Assurance Representative
QDR	Quality Deficiency Record
QQPRI	Quantitative and Qualitative Personnel Requirements Information
RADC	Rome Air Development Center
RAM	Random Access Memory
RJE	Remote Job Entry
RFP	Request for Proposal
ROC	Required Operational Capability
ROM	Read-Only Memory
RPG	Report Program Generator
SAMSO	Space And Missile Systems Organization
SAMSOP	Space And Missile Systems Organization Pamphlet
SCN	Specification Change Notice
SDR	System Design Review
SOW	Statement of Work
SPO	System Program Office
SQA	Software Quality Assurance
SRR	System Requirements Review
S/W	Software
TAC	Tactical Air Command
TCTO	Time Compliance Technical Order
TDD	Top-Down Design
TDI	Top-Down Implementation
T&E	Test and Evaluation

TEMP	Test and Evaluation Master Plan
TEPI	Training Equipment Planning Information
TEOA	Test and Evaluation Objectives Annex
TI	Technical Interchange
TTY	Teletypewriter
VDD	Version Description Document
V&V	Validation/Verification
WBS	Work Breakdown Structure
WWMCCS	World Wide Military Command and Control System

HANDBOOK GLOSSARY

The intent of this glossary is to assist personnel in the use of this handbook only.

ABSOLUTE ADDRESS - A memory address/location in a computer's storage that is specifically identified. See also **BASE ADDRESS** and **RELATIVE ADDRESS**.

ABSOLUTE ADDRESSING - Utilizing absolute addresses in an instruction.

ABSOLUTE CODING - Coding/writing computer instructions that include absolute addresses.

ACCEPTABILITY - The degree to which software or hardware meets the needs of a user or the effectiveness of the man-machine interface.

ACCESS TIME - The time elapsed between the moment a request is made for data from storage until the moment the data is received.

ACCUMULATOR - A register in the arithmetic unit of the computer which can be used to store the results of an operation.

ACCURACY - The quality by which systems, programs, or data are measured to be error free.

ACRONYM - A term developed from the first letters or selection of letters in a group of related words, such as ESD for Electronic Systems Division.

ADAPTABILITY - The ease with which software or hardware can be changed to meet new or revised user requirements or changing system environments.

ADDRESS - Character(s) that identify a location in a storage device.

ALGOL - An acronym for **ALGORITHMIC LANGUAGE**. A high-level computer language designed to present computer algorithms in a generally understood and strictly procedure-oriented manner.

ALGORITHM - A definitized set of rules for solving a problem in a finite number of steps.

ALGORITHMIC LANGUAGE - A language which can present computer programs by algorithms.

ALLOCATED BASELINE - The configuration identification established at the end of the requirements/performance phase.

ALPHANUMERIC - Pertaining to a set of alphabetic, numeric, and symbolic characters in some combination.

ANALOG COMPUTER - A computer that performs continuously varying physical processing of data represented by nondiscrete values. Contrast to DIGITAL COMPUTER.

ANALOGY - Inference that if two or more items are similar in some respects, they may probably agree in others.

ARRAY PROCESSORS - A system composed of identical processing units functioning under the control of a master CPU. Same as SINGLE INSTRUCTION MULTIPLE DATA STREAM COMPUTERS.

ASSEMBLE - To create an object language program from a source language program by substituting machine-oriented instructions for symbolic machine language instructions.

ASSEMBLER - A computer program used to translate source coded instructions into object coded instructions. See ASSEMBLE.

ASSOCIATIVE MEMORY - A storage device in which locations are identified by their contents rather than by names or addresses.

ASYNCHRONOUS COMPUTER - A computer in which each operation commences as a result of a signal generated by the completion of a previous operation. Contrast with SYNCHRONOUS COMPUTER.

AUTOMATIC DATA PROCESSING (ADP) - The processing of data by automatic means with the minimum of human intervention.

AUTOMATIC DATA PROCESSING SYSTEM (ADPS) - An assembly of automatic data processing resources including equipment, procedures, and communications, as well as personnel in specified cases.

AUTOMATIC PROGRAMMING - Utilizing a computer to assist with certain segments of program development.

AUXILIARY STORAGE - Computer memory or storage other than main memory, such as tape or disc. Same as SECONDARY STORAGE.

AVAILABILITY - The probability that a computer system is operating satisfactorily at a specific time and has the flexibility and capacity to accept an additional workload.

BACKGROUND PROCESSING - The machine execution of lower priority computer programs when higher priority programs are not utilizing system resources. Also see FOREGROUND PROCESSING.

BACKUP - Pertaining to additional or reserve system resources that may be utilized in the event of problems with the normally used sources.

BASE ADDRESS - The numeric value used in combination with the numeric value of the relative address to develop the absolute address. Base address plus relative address equals absolute address. See also **RELATIVE ADDRESS** and **ABSOLUTE ADDRESS**.

BASELINES - A configuration identification document or set of such documents, and also the computer system or program itself after the product baseline, formally designated and fixed at a specific time during the acquisition life cycle. Baselines, plus approved changes to those baselines, constitute the current configuration identification.

BASIC - An acronym for Beginner's All-Purpose Symbolic Instruction Code. A simple, easy to learn high-level language.

BATCH PROCESSING - The processing of data accumulated over a period of time or grouped by job. Same as **SEQUENTIAL PROCESSING**.

BAUD - A unit of measurement for signaling speed equal to the number of code elements per second.

BENCHMARKING - A process of applying a test workload on one or more components of a computer system, or one or more computer systems to determine specific capabilities.

BINARY-CODED DECIMAL (BCD) - A binary-coded notation consisting of decimal digits, made up of four binary numerals.

BINARY CODE - A code made up of two discrete characters, usually 0 and 1.

BINARY DIGIT - See **BIT**.

BIT - An acronym for Binary Digit, that is the most elementary unit of information in digital computing, either of the digits 0 or 1.

BLOCK - A grouping of characters, words or records that are processed or treated as an entity.

BLOCK LENGTH - The number of characters, words or records contained in a block of data.

BOOTSTRAP LOADER - An input routine used to initiate the loading of an entire program of which it is a part.

BOTTOMS-UP DESIGN - A traditional procedure of software development where the lowest-level processing programs are coded first, module-tested, and made ready for integration with additional programs. Work proceeds in this manner up the hierarchy of the design.

BRANCH - An alternative route in a logic flow.

BREAKPOINT - A point at which a program may be interrupted for a variety of purposes.

BUFFER - A routine or storage device used to balance data rate flow differences which occur with the movement of data between various devices of a computer system.

BUG - An error or flaw in a program or equipment component.

BULK STORAGE - Large auxiliary storage or memory. Same as MASS STORAGE.

BUS - A circuit or path used for data or power transfers in a computer system.

BYTE - A physical or logical grouping of binary digits usually operated upon as a unit.

CACHE MEMORY - A small, high speed storage device which is imposed between a processor and the main memory in a computer system to improve the system's performance. Also see SEMICONDUCTOR MEMORY.

CATHODE RAY TUBE (CRT) DISPLAY - An input/output device which provides visual displays of data, instructions, procedures, etc., and which usually provides for user input from a keyboard type of device.

CENTRAL PROCESSING UNIT (CPU) - The unit of a computer system which controls the interpretation and execution of program instructions. Same as CENTRAL PROCESSOR and MAIN FRAME.

CENTRAL PROCESSOR - See CENTRAL PROCESSING UNIT (CPU).

CHANNEL - A path along which data is transferred between various units or components of a computer system.

CHARACTER - A letter, digit or symbol representing data.

CHARACTER RECOGNITION - The reading and encoding of characters by automatic means.

CHECK BIT - See PARITY BIT.

CHECK DIGIT - A digit used to check for the absence of errors.

CLARITY - A measure of human effort required to comprehend a program or its documentation.

CLOCK - A device that measures time, indicates time, or generates periodic signals in a synchronous computer to control timing of operations.

COBOL - An acronym for COMmon Business Oriented Language. A high-level language designed for commercial data-processing problems.

CODE - Representations or symbols for data, characters, text, etc.

CODING - The process of preparing a part or all of a program in a high-level or machine-oriented language.

COLLATE - To change the arrangement of a group of data or items.

COMMAND AND CONTROL (C2) APPLICATION - The automated on-board decision making processes that direct the operations of an activity or vehicle.

COMMAND, CONTROL AND COMMUNICATIONS (C3) SYSTEMS, AIR FORCE - Encompasses those systems required by the Air Force to accomplish its defense, surveillance, and offense mission responsibilities. Such systems may be ground based (fixed and mobile), airborne, or spaceborne; or any combination thereof. Systems may also be categorized as strategic, tactical, or air defense systems.

COMMUNICATION APPLICATION - The automated process of transmitting data from one location to another.

COMPATIBILITY - A measure of interoperability that can be expected of software or hardware with other software or hardware units.

COMPILE - To prepare an object or machine-oriented language program from a program written in a high-level language.

COMPILER - A computer program designed to compile a source language into an object language. Same as COMPILING PROGRAM. Compare to ASSEMBLER.

COMPILING PROGRAM - See COMPILER.

COMPLETENESS - Those attributes of software or hardware that provide full implementation of the functions required. Also see CORRECTNESS.

COMPLEXITY - The degree of complication of a software product, consisting of the weighted factor of such measures as the number of control paths, number of shared data references, number of loops, number of interactions between system components, user interfaces, and hardware.

COMPUTER - A data processor that can perform a multitude of arithmetic and logic computations with the minimum of human intervention.

COMPUTER INSTRUCTION - An instruction designed to be recognized by the Central Processing Unit of a computer. Same as MACHINE INSTRUCTION.

COMPUTER NETWORK - Two or more computers interconnected through communication links. Same as NETWORK.

COMPUTER-ORIENTED LANGUAGE - See MACHINE-ORIENTED LANGUAGE.

COMPUTER PROGRAM - A series of instructions or statements in a form that may directly or indirectly be acceptable to a data processing system. Also see SOFTWARE.

COMPUTER PROGRAM COMPONENT (CPC) - A functionally or logically distinct part of a computer program distinguished for purposes of convenience in designing and specifying a complex computer program as an assembly of subordinate elements. Also see FUNCTION.

COMPUTER PROGRAM CONFIGURATION ITEM (CPCI) - A specification that defines the requirements for an end item of a computer program.

COMPUTER PROGRAM DEVELOPMENT SPECIFICATIONS - A document which specifies the total functional performance requirements for each CPCI. A specification that represents a comprehensive and definitive statement of the performance, design, and test requirements to be met by a computer program. Equivalent to "Part I CPCI specification" or "Type B5 specifications".

COMPUTER PROGRAM PRODUCT SPECIFICATION - A document or series of documents which contain the detailed technical description of the Computer Program Configuration Item (CPCI) as designed and coded. It is a complete description of all routines, limits, timing, flow, and coded instructions. Equivalent to "Part II CPCI specification" or "Type C5 specification".

COMPUTER RESOURCES - The totality of computer equipment, computer programs, computer data, associated documentation, personnel, and supplies. Also see EMBEDDED COMPUTER RESOURCES.

COMPUTER SOFTWARE - A combination of associated computer programs and computer data required to enable the computer equipment to perform data processing or control functions. Also see SOFTWARE.

COMPUTER SYSTEM - See AUTOMATIC DATA PROCESSING SYSTEM.

CONCISENESS - The ability to satisfy functional requirements with a minimum amount of software or hardware.

CONDITIONAL BRANCH - A logic flow transfer action which will only occur if certain specified conditions exist when the program statement is executed. Also see UNCONDITIONAL BRANCH.

CONFIGURATION - The functional and/or physical characteristics of software or hardware as set forth in technical documentation and achieved in a product.

CONFIGURATION CONTROL - The systematic evaluation, coordination, approval or disapproval, and implementation of all approved changes in the configuration of a Configuration Item (CI) after formal establishment of its configuration identification.

CONFIGURATION ITEM (CI) - An aggregation of software or hardware, or any of its discrete portions, which satisfies an end-use function and is designated by the Government for configuration management. CIs may vary widely in complexity, size, and type, from a C3 system to a test meter. Also see CRITICAL ITEM.

CONFIGURATION MANAGEMENT - The application of technical and administrative management direction and surveillance to accomplish: the identification, authentication, and recording of the functional and physical characteristics of a system; the control of changes to identified and authenticated characteristics; and the maintenance of records and issuance of reports on configuration status.

CONSISTENCY - The degree to which software or hardware satisfies specifications, or the extent that it contains uniform notation, terminology, and symbology within itself, and the extent that the content is traceable to the requirements.

CONSOLE - A unit of a computer used for communication between a computer and a user, usually a system operator or manager.

CONTRACT DATA REQUIREMENTS LIST (CDRL) - A contract form, DD 1423, listing all technical data items and status data items, selected from an Authorized Data List, to be delivered under a contract.

CONTROL FUNCTION - The automated procedures involved in initiating an action or process, and subsequently adjusting that action or process based on feedback data.

CONVERSATIONAL MODE - A mode of operation of a computer system input/output device which allows a dialog between a user and a system during the execution of a program.

CONVERTER - A device that converts data from one storage media to another, such as from tape to disc.

CORE SIZE - The storage capacity of a computer magnetic core main memory, which indicates the maximum number of words, of a designated length, that can be stored at one time.

CORE STORAGE - A nonmoving magnetic storage unit that records data or programs by setting the direction of magnetization in small toroidal shaped magnetic material. Reading magnetic core erases the information so that each read must be followed by a write cycle. Same as MAGNETIC CORE STORAGE.

CORRECTNESS - The property of performing as intended for all acceptable inputs, and the extent to which software or hardware satisfies its specifications and fulfills the user's mission objectives. Also see **COMPLETENESS**.

CRITICAL DESIGN REVIEW (CDR) (COMPUTER PROGRAM) - A formal technical review of the design as depicted by the specification and flow diagrams, sufficiently detailed to enable a programmer to code, compile, and debug a computer program, to assure that design requirements have been met before beginning coding.

CRITICAL DESIGN REVIEW (CDR) (HARDWARE) - A formal technical review of the design of a machine item to assure that design requirements have been met.

CRITICAL ITEM - An item within a configuration item (CI) which, because of special engineering or logistic considerations, requires an approved specification to establish technical or inventory control at a level below the CI level. The critical item designation does not apply to computer programs. Also see **CONFIGURATION ITEM**.

CYBERNETICS - The study of theories and concepts of control and communications in and between living organisms and machines.

CYCLE TIME - The minimum time interval between the starts of successive read/write cycles of core storage.

DATA ACQUISITION SYSTEM - A system which gathers performance data on a computer program during stages of development and testing, as it performs in a computer.

DATA BASE - A collection of data essential to a computer system's operation.

DATA BASE MANAGEMENT SYSTEM (DBMS) - A software system which controls the access, storage, updating, and maintenance of a data base.

DATA ITEM DESCRIPTION (DID) - A standard form (DD Form 1664) employed to define format and content requirements for specifications, reports, manuals, and various other items of technical or management data to be delivered under a contract.

DATA REDUCTION - The process of changing raw data into a usable form.

DEBUG - To locate and eliminate errors in a program or malfunctions in computer system.

DECISION TABLE - A table of contingencies related to a specific problem, together with the corresponding actions to be taken.

DELAY LINE - A line or network device designed to reduce the speed at which signal transmissions are made. Historically used as the main memory device of computers.

DEPENDABILITY - The probability that software and/or hardware will perform in its intended environment.

DEVELOPMENT PROCESS - A formal process by which software or hardware requirements are transformed into design specifications, created, or built and placed in operational status.

DIGITAL COMPUTER - A computer that performs arithmetic and logical operations on data which is represented by discrete values. Contrast with ANALOG COMPUTER.

DIRECT ACCESS - The ability to obtain or enter data directly from or to a computer storage location without any prior data reference. Same as RANDOM ACCESS. Contrast with HIERARCHIAL MEMORY.

DISC STORAGE - A rotating magnetized disc which can store data or programs. Same as MAGNETIC DISC STORAGE.

DISK - Variation of DISC.

DISPLAY FUNCTION - The automated procedures involved in producing the resultant type and format of the input process and the output process of a system.

DOCUMENTATION - Publications which describe the design, logic, and operations of software or hardware.

DRUM STORAGE - A revolving metal cylinder covered with a magnetic sensitive surface which can store data or programs. Same as MAGNETIC DRUM.

DUMP - To output/print the contents, data or programs, stored in a storage device.

DUPLEX - In data communications, the capability to simultaneously transmit in two different directions over a single channel.

DYNAMIC STORAGE ALLOCATION - A technique for allocating computer storage space for data and programs on a controlled, selective basis. A way that at one time an item may occupy one part of storage and at another time it may occupy another, depending on the circumstances.

EDIT - To test, correct or modify data in preparation for a subsequent operation.

EFFICIENCY - The degree to which a task is performed with a minimum consumption of time and resources. In a computer, obtaining maximum throughput with minimum execution time, storage space, and peripheral device utilization. Also see **EXECUTION EFFICIENCY**.

EMBEDDED COMPUTER SYSTEM(S) (ECS) - An ECS is integral to an electronic or electromechanical system such as in command, control, and communications systems, combat weapon system, tactical system, aircraft, ship, missile, or spacecraft from a design, procurement, and operations viewpoint. Key attributes include being physically incorporated into a larger system whose function is not data processing; integral to, or supportive of, a larger system from a design, procurement and operations viewpoint; and outputs include information, control signals, and computer data.

EMBEDDED COMPUTER RESOURCES - The totality of computer equipments, programs, data, and communication links within a larger system not dedicated to data processing. Also see **COMPUTER RESOURCES**.

EMULATE - To imitate one computer system by another computer system so that the second system will accept data and programs intended for the first system.

ENVIRONMENT - The conditions or circumstances surrounding and influencing the operation of a computer system.

ERLANG - A unit of measurement for traffic intensity, measured as the total number of signals or messages received during a mean service time.

EXECUTION EFFICIENCY - Those attributes of the software or hardware that provide for minimum processing time. Also see **EFFICIENCY**.

EXECUTIVE SYSTEM - See **OPERATING SYSTEM**.

EXPANDABILITY - Those attributes of the software that provide for expansion of data storage requirements or computational functions. Also see **FLEXIBILITY**.

EXTENSIBILITY - Extent to which software or hardware can support extensions of critical functions.

FALL-BACK PROCEDURES - Programs that operate in such a way as to circumvent a fault that occurs in a computer system, and which may or may not give a degraded service.

FETCH - To obtain data from storage.

FIELD - A part of a data record.

FILE - A collection of related data records considered as a unit.

FILE MAINTENANCE - The process of updating data maintained in file.

FIRING SEQUENCE CONTROL FUNCTION - The automated procedures involved in directing the pattern of ignitions of missiles, rockets, satellite directional jets, etc.

FIRMWARE - Computer programs and data loaded in a class of memory that cannot be dynamically modified by the computer during processing.

FLEXIBILITY - Extent to which a system can absorb workload increases and decreases, or the ability of a system to immediately handle different logical situations. Also see EXPANDABILITY.

FLOWCHART - A logic diagram which illustrates the sequential processing steps of a computer program or computer system. Same as SYSTEM FLOWCHART.

FOREGROUND PROCESSING - The machine execution of high priority computer programs which preempts the use of system resources by lower priority programs. Also see BACKGROUND PROCESSING.

FORMAL QUALIFICATION REVIEW (FQR) - A review that normally occurs at completion of software validation testing to certify that the test results correspond to preestablished acceptance criteria. Successful completion of the FQR establishes the Product Baseline.

FORMAL QUALIFICATION TEST (FQT) - That portion of software testing which is conducted in accordance with approved test plans for the purpose of verifying that the software fulfills its requirements. The FQT is a complete and comprehensive test in a continuous test period prior to Functional Configuration Audit (FCA). Also see FUNCTIONAL CONFIGURATION AUDIT.

FORTRAN - An acronym for FORMula TRANslator. A high-level language designed to facilitate the performance of mathematical computations.

FREQUENCY OF OPERATION - Measure of average time between executions of a program.

FUNCTION - A specific purpose of an entity of hardware or software such as what action it will direct or perform. Also see COMPUTER PROGRAM COMPONENT.

FUNCTIONAL BASELINE - The initial system configuration identification established at the end of the conceptual phase, normally existing prior to the start of a software development project.

FUNCTIONAL CONFIGURATION AUDIT (FCA) - A formal examination of the test data for a Configuration Item's functional characteristics prior to acceptance, to verify that the item has achieved the performance specified by its functional or allocated configuration identification. Also see FORMAL QUALIFICATION TEST.

GENERATE - To create a machine-oriented language program from a selection of parameters.

GENERATOR - A computer program that performs a generate function. See GENERATE.

GROSCH'S LAW - Predicts that computing power increases as a factor of the square of the cost.

HARDWARE - The electric, electronic, and mechanical equipment used for processing data consisting of cabinets, racks, transistors, wires, motors, and such; or any component of automatic data processing equipment.

HARDWARE ROUTINES - Integrated circuit logic units that provide machine instructions for basic functions such as multiply, divide or fronting-point conversion without the need for software development.

HEURISTIC METHOD - An exploratory method of solving problems by trial and error.

HEXADECIMAL NUMBER SYSTEM - A base 16 number system which uses the digits 0 through 9, and the letters A through F as its symbols.

HIERARCHIAL MEMORY - A logical memory structure that orders the programs data and instructions so that one leads to the next within a block of memory. Contrast with RANDOM-ACCESS MEMORY.

HIGH-LEVEL LANGUAGE (HLL) - A machine independent language designed for programming ease; must be compiled for use with a specific computer. Also see SOURCE LANGUAGE. Same as HIGHER-ORDER LANGUAGE.

HIGHER-ORDER LANGUAGE (HOL) - See HIGH-LEVEL LANGUAGE (HLL).

HOLLERITH CODE - A type of code initially developed by Hollerith for use in Electrical Accounting Machine (EAM) cards.

HOUSEKEEPING OPERATIONS - Operations that assist software or hardware in accomplishing its functions, yet are not actually a part the software or hardware. Same as OVERHEAD OPERATIONS.

IMMEDIATE ADDRESS - The portion of an address that contains the operand.

INDIRECT ADDRESS - An address that identifies the location of data to be treated as the address of an operand.

INPUT/OUTPUT (I/O) - Pertaining to a device which performs the input process and the output process of a computer system.

INSTRUCTION - A unit of logic in a computer program that specifies one operation and its operands.

INTEGRATED DATA PROCESSING (IDP) - Data processing that incorporates data acquisition and data processing into a single system.

INTEGRITY - The ability of one software or hardware subsystem to protect the operation of another, or a measure of the degree of protection a software or hardware subsystem or systems offers against unauthorized access and loss due to controllable events.

INTERFACE - A common boundary between units of a computer system, between computer systems, or between a computer system and another system.

INTERFACING APPLICATION - The automated processes that control the interchange of data between units of a system or between systems.

INTERLEAVING - A processing technique of alternating between parts of two or more programs, data, or events while maintaining the identity of each.

INTERPRET - To translate and execute source language statements, one at a time, in sequence.

INTERRUPT - To suspend an ongoing process in order to accomplish a pre-determined function and subsequently may allow the resumption of the original ongoing process. Often signalled by some specific condition such as input/output completion, hardware errors, or some timing function.

ITEM - A portion of data grouping that is considered as a unit.

JOB - A set of data upon which a computer operates that is considered a unit of work.

JOB CONTROL LANGUAGE (JCL) - A language designed to identify a job and the job's requirements upon an operating system.

LABEL - The identification of a set of data or in a program, an identifier of an instruction.

LASER - An acronym for Light Amplification by Stimulated Emission of Radiation; a device that produces an intense, coherent, directional beam of light.

LASER READ-ONLY MEMORY - A device that uses a low-power laser to read a data pattern in a light sensitive film that has been previously recorded by a high power laser.

LIBRARY - A collection of related data files or programs; or a storage area for magnetic tapes or disc packs.

LOOKAHEAD - A form of processing in which an instruction is fetched and prepared for execution while a previous instruction is still being executed.

LOOP - A set of instructions that under specified circumstances will be executed repeatedly.

MACHINE INSTRUCTION - See **COMPUTER INSTRUCTION**.

MACHINE-ORIENTED LANGUAGE - A language designed or designated for a specific type or class of computers. Same as **COMPUTER-ORIENTED LANGUAGE**.

MACROINSTRUCTION - A source language instruction that is replaced by a set of defined source language instructions.

MAGNETIC CORE STORAGE - See **CORE STORAGE**.

MAGNETIC DISC STORAGE - See **DISC STORAGE**.

MAGNETIC DRUM - See **DRUM STORAGE**.

MAGNETIC TAPE STORAGE - A tape that is coated on one side with a magnetic sensitive surface to store data or programs. Same as **TAPE STORAGE**.

MAIN FRAME - See **CENTRAL PROCESSING UNIT (CPU)**.

MAIN MEMORY - See **MAIN STORAGE**.

MAIN STORAGE - The primary memory or storage device in a computer system associated with the central processing unit (CPU).

MAINTAINABILITY - The extent to which a software product facilitates updating to correct errors and to satisfy new requirements. A maintainable software product is one which is understandable and testable and can be easily modified to rectify a deficiency and/or add new capabilities.

MAINTENANCE - Changes, modifications, restructuring or recoding of the software for whatever reason, and is used synonymously with software support.

MANAGEABILITY - The degree to which a computer system lends itself to efficient administration of its components.

MANAGEMENT INFORMATION SYSTEM (MIS) - An automated information system designed to assist management in decision making.

MAP - A listing of data and programs with their related locations in a memory device.

MARK SENSING - The automatic sensing of manually made marks in an input document.

MASS STORAGE - See BULK STORAGE.

MASTER FILE - A relatively permanent file of data for a specific job which can be updated if required.

MEAN-TIME-BETWEEN-FAILURES (MTBF) - A determined average period of time, under specified conditions, that a functional unit will not fail in an assumed life of a unit.

MEAN-TIME-TO-REPAIR (MTTR) - A determined average period of time to accomplish the repair of a functional unit in an assumed life of a unit.

MEGABIT - One million binary bits.

MEGABYTE - One million bytes.

MERGE - To combine two or more sets of items into one distinct set, usually in some logical order.

MICROPROGRAM - A sequence of instructions, hardwired in a computer, which the computer uses to interpret machine language instructions.

MICROSECOND - One millionth of a second, represented as μ s or microsec.

MILLISECOND - One thousandth of a second, represented as ms or msec.

MISSILE FIRE CONTROL APPLICATION - The automated processes that direct the launching sequence for a missile.

MODEM - An interface device that functions as a modulator and demodulator between a computer system and a communication link to another system, unit, device or sensor.

MODIFIABILITY - A quality of software or hardware that reduces the effort required to alter it in order to conform to a modification of its specification.

MODULE - A set of source instructions in a form consistent with the appropriate language, and computer system that encompasses one specific function and has only one entry statement and one exit statement. At ESD, a module should not exceed 100 lines of executable source code, excluding comments and data definitions.

MONITOR - A device or set of routines that observe a data processing system's operation and identifies functions occurring, probable problem areas, or system performance.

MULTIPLEX - To simultaneously use a single channel of a communications link to transmit two or more messages.

MULTIPROCESSOR - A computer system with two or more central processing units functioning under integrated control.

MULTIPROGRAMMING - The simultaneous execution of two or more programs by a central processing unit, usually effected by interleaving the programs execution under control of an operating system which attempts to optimize overall performance..

NANOSECOND - One billionth of a second, represented as ns or nanosec.

NAVIGATION APPLICATION AND FUNCTION - The automated processes involved in determining the spatial position of a vehicle, such as an aircraft, missile, satellite, etc., and accomplishing the necessary computations to direct the vehicle towards another position.

NETWORK - See COMPUTER NETWORK.

OBJECT CODE - The output machine language from an assembler or compiler. Same as OBJECT LANGUAGE.

OBJECT LANGUAGE - See OBJECT CODE.

OBJECT PROGRAM - A program assembled or compiled in object code.

OCTAL NUMBER SYSTEM - A base 8 number system which uses the digits 0 through 7 as its symbols.

OFF-LINE - Pertaining to devices or equipment that are not under the direct control of the central processing unit.

OFF-LINE STORAGE - Storage that is not under the direct control of the central processing unit.

ON-LINE - Pertaining to devices or equipment that are under the direct control of a central processing unit.

ONLINE PROCESSING - See ON-LINE.

OPERAND - Data or an address upon which an operation is applied.

OPERATING SYSTEM (O/S) - Computer software that directs the execution of computer programs and in some instances may also supervise functions, such as accounting, assigning storage, compiling, controlling I/Os, data managing, debugging, and others. Same as EXECUTIVE SYSTEM.

OPTICAL CHARACTER RECOGNITION (OCR) - A technique of using a light-sensitive device to read printed input data.

OVERFLOW - The portion of data that exceeds the prescribed length or limits of a storage location.

OVERHEAD OPERATIONS - See HOUSEKEEPING OPERATIONS.

OVERLAY - The technique of utilizing the same areas of memory repeatedly during various stages of the data processing operation. This technique makes it possible to execute programs that are too large to fit into the main memory of a computer.

PACK - To place data in a compact form into memory by employing certain characteristics of the data and memory. Also see UNPACK.

PAGE - A group of data or instructions, or both, contained in a computer's memory, and usually a specific size fixed by hardware design.

PAGING - A time sharing technique in which blocks of instructions or data are transferred with a computer system.

PARALLEL - Pertaining to concurrent operations of two or more entities. Same as PARALLEL PROCESSING.

PARALLEL COMPUTER - A computer that can perform concurrent or parallel operations. Also see SERIAL COMPUTER.

PARALLEL PROCESSING - See PARALLEL.

PARAMETER MEASUREMENT FUNCTION - The automated procedures involved in determining distances, or quantities such as capacity, volume, time, speed, pressure, temperature, pulse rate, etc.

PARITY BIT - A binary control digit attached to a group of binary digits so that the resultant total is either odd or even, dependent upon established conditions of a particular system. Same as CHECK BIT.

PARITY CHECK - A test check to determine if the number of ones or zeros in a group of binary digits is either odd or even to determine if a single bit has been changed.

PASS - One complete cycle of processing a specified amount or group of data.

PATCH - To quickly or temporarily correct a program or computer system in order that it may resume functioning.

PERFORMANCE - Pertaining to ability of a computer system or subsystem to perform its functions, measured in such terms as response time, throughput, and turnaround time. These measures quantify the performance of the system or subsystem with respect to time versus workload.

PERIPHERAL EQUIPMENT - Computer system equipment, other than the central processing unit and/or main memory.

PHYSICAL CONFIGURATION AUDIT (PCA) - The formal examination of the coded configuration of a program element against its technical documentation in order to establish the element's initial configuration identification.

PICOSECOND - One trillionth of a second, represented by ps or psec.

PIPELINING - A form of parallel processing in which a class of instructions may be simultaneously executed in different stages within a processing unit.

PL/I - See PROGRAMMING LANGUAGE I.

PORTABILITY - The ability to readily transfer a program from one computer system and/or software system environment to another.

PRELIMINARY DESIGN REVIEW (PDR) - A review that normally occurs at completion of the System Design Phase. Successful completion of this review establishes the preliminary computer system development specifications, interface specifications, and data requirements specifications in the System Design Baseline.

PROBLEM-ORIENTED LANGUAGE - A high-level language designed for problem solving such as procedure-oriented languages or simulation languages.

PROCEDURE-ORIENTED LANGUAGE - A high-level language designed to accommodate easy development of algorithmic procedures.

PROCESSOR - A device of a system capable of performing operations on data. May be either hardware or software.

PRODUCT BASELINE - The configuration identification established at end of the test and acceptance phase.

PROGRAM - A set of procedures for accomplishing solutions for particular problems.

PROGRAMMING FLOWCHART - See FLOWCHART.

PROGRAMMING LANGUAGE I (PL/I) - A high-level language designed for business and scientific applications.

QUEJING THEORY - A field of probability theory useful in analyzing delays at critical points or nodes in a process or some device configuration.

RANDOM ACCESS - See DIRECT ACCESS.

READ-ONLY MEMORY (ROM) - A storage device which contains data or programs that can not be inadvertently erased or overwritten during the normal operations. Same as SPECIAL-PURPOSE MEMORY. See SEMICONDUCTOR MEMORY.

REAL-TIME - Pertaining to the accomplishment of a processing operation during the same time a related physical process is occurring and which the processing operation can influence. Same as REAL-TIME PROCESSING, REAL-TIME SYSTEM.

REAL-TIME PROCESSING - See REAL-TIME.

REAL-TIME SYSTEM - See REAL-TIME.

RECORD - A group of related data which is treated as a unit.

RECORDING DENSITY - The number of bits in a single track measured per unit of length of a recording medium.

REGISTER - A special purpose storage device, associated with the central processing unit, for storing specified data.

RELATIVE ADDRESS - The numeric value used in combination with the numeric value of the base address to develop the absolute address. See also ABSOLUTE ADDRESS and BASE ADDRESS.

RELIABILITY - The probability that software or hardware will perform a required function under specific conditions, without failure, for a specified period of time.

REPORT GENERATOR - A generator which formats reports based on design parameters.

RESPONSE TIME - The elapsed time between the end of a query input until the start of response output for a given workload.

REUSABILITY - Extent to which software or hardware can be used in other applications or operations.

ROBUSTNESS - Extent to which software or hardware will continue to perform despite some violations of the basic assumptions in its specifications.

ROUTINE - A set of instructions that define a process.

RUN - The performance or accomplishment of a job or operation.

SECONDARY STORAGE - See AUXILIARY STORAGE.

SEMICONDUCTOR MEMORY - A solid state large-scale integrated circuit that has individual circuits which can be set in a conducting or nonconducting state; used in high speed buffers, read-only memory, and microprogrammable memory. See CACHE MEMORY, READ-ONLY MEMORY. Same as LSI MEMORY.

SEQUENTIAL PROCESSING - See BATCH PROCESSING.

SERIAL ACCESS - The ability to obtain or enter data into a computer storage device only in a sequential manner.

SERIAL COMPUTER - A computer which can only perform operations sequentially. Also see PARALLEL COMPUTER.

SETUP TIME - The time required to prepare a computer system for a specified processing operation.

SEXADECIMAL - See HEXADECIMAL.

SIGNAL PROCESSING APPLICATION - The automated processes involved in manipulating data prior to transmission or subsequent to receipt.

SINGLE INSTRUCTION MULTIPLE DATA STREAM COMPUTERS - See **ARRAY PROCESSORS**.

SIZE, COMPUTER SYSTEM - The size of an embedded computer system has two components: 1) physical size; in terms of the number and types of physical hardware units and the number of unique software entities including data and their corresponding number of source statements, object words, or required storage space as appropriate; and 2) capacity; in terms of storage or processing capability of hardware which is not dependent upon software.

SIZING, COMPUTER SYSTEM - Activities involved in estimating the physical and functional (configuration) aspects of a computer system's components such as core memory, auxiliary memory, software, virtual memory, and I/O devices, to determine its performance capabilities for processing data.

SNAPSHOT DUMP - An output report of a selected storage area taken at particular points in time of a program's execution.

SOFTWARE - Computer programs and in certain cases, associated documentation. The two types of software are: (1) basic software, which consists of programs designed to facilitate the use of a particular computer system and as an operating system (O/S) or data base management system (DBMS); and (2) application software, which consists of programs designed by or for computer system users to accomplish specific data processing tasks such as command, control, and communications, weapons control, etc.

SOFTWARE ENGINEERING - The science of design, development, implementation, test, evaluation, and maintenance of computer software over its life cycle.

SOLID STATE COMPONENT - A component designed in a solid physical state such as a transistor.

SOURCE CODE - See **SOURCE LANGUAGE**.

SOURCE LANGUAGE - A computer program written in a symbolic language designed for programming ease which must be translated into a machine-oriented language to enable it to be machine processable. Also see **HIGH-LEVEL LANGUAGE (HLL)**, **HIGHER-ORDER LANGUAGE (HOL)**, **SOURCE PROGRAM**. Same as **SOURCE CODE**.

SOURCE PROGRAM - A program written in a high-level language. Also see **SOURCE LANGUAGE**.

SPECIAL-PURPOSE COMPUTER - A computer designed for special problems or environments.

SPECIAL-PURPOSE MEMORY - See **READ-ONLY MEMORY (ROM)**.

STORAGE - A storage device or the capacity to hold data or programs in a computer system either temporarily or permanently.

STORAGE ALLOCATION - The assignment of storage areas for designated data or programs.

STORAGE PROTECTION - Protection built into a computer system which precludes reading and/or writing access to a designated storage area.

STRUCTURED PROGRAMMING - A programming technique, based on the mathematically proven Structure Theorem, which utilizes top-down program development, programming support libraries, and chief programmer team concepts.

SUBROUTINE - A portion of a computer program routine that performs a specific or generalized function.

SYNCHRONOUS COMPUTER - A computer in which each operation commences with predetermined signals from a clock. Contrast with **ASYNCHRONOUS COMPUTER**.

SYSTEM DESIGN REVIEW (SDR) - A review that normally occurs on completion of the System Design Phase. Successful completion of this review establishes the preliminary or system-level computer system development specifications, interface specifications, and requirements specifications in the System Design Baseline.

SYSTEM FLOWCHART - See **FLOWCHART**.

SYSTEM MONITORING FUNCTION - The automated procedures that evaluate the operational and functional status of a system.

TABLE - A collection of data arranged in such a manner to facilitate easy reference during data processing operations, such as a table of terms or values.

TABLE LOOK-UP - The act of obtaining a specified data item in a table.

TAPE STORAGE - See **MAGNETIC TAPE STORAGE**.

TARGET DATA ENTRY FUNCTION - The automated procedures involved in the conversion of signals received by radar or sensors into a form capable of being processed by an Embedded Computer System (ECS).

TARGET IDENTIFICATION FUNCTION - The automated procedures involved in processing data to determine if a target is friendly or hostile.

TARGET TRACKING FUNCTION - The automated procedures involved in determining the spatial position of one or more moving entities at regular intervals in order to calculate distance, course, speed, etc.

TELECOMMUNICATIONS - The transmission of messages, data or signals by electronic means.

TELETYPEWRITER (TTY) - A typewriter-like device used to send or receive messages.

TESTABILITY - Effort required to test software or hardware to insure it performs its intended function.

THIN FILM MEMORY - A storage device that utilizes a magnetic film, on a thin glass plating, which is polarized for data storage.

THROUGHPUT, COMPUTER SYSTEM - 1) The total production or workload of a computer system from initial input of data, through processing, and finally to the output of data; or 2) the work completed per unit time for a given workload.

TIME SHARING - A mode of operation that provides for more than one user to utilize a single computer system or units thereof.

TIMING, COMPUTER SYSTEM - The activities involved in estimating the speed at which a computer system can handle a data processing function. The timing of a computer system has two components: 1) workload independent parameters such as memory access time, cycle time, printer output rate, etc., and 2) workload dependent measures such as throughput, response time, and utilization.

TOLERANCE - Measure of the ability of a computer system to accept different variations of the same data as valid or withstand a degree of variation in input without malfunctions or rejections.

TOP-DOWN DESIGN (TDD) - The concept of hierarchical design encompassing tiered levels, specifications, and modules subordinate to the overall or total system level, such as observed in tree diagrams, break-down structure, and top-down structured programming.

TOP-DOWN IMPLEMENTATION (TDI) - The technique of implementing software down a hierarchical structure which facilitates early testing of selected segments.

TRACE - A record of how computer program instructions were executed.

TRACEABILITY - Those attributes of software that provide continuity from the requirements to the implementation.

TRANSLATOR - A device or computer program that translates programs in one language to another program language.

TUNING - Making minor modifications to system's hardware, software or other aspects of a data processing operation for the purpose of increasing the efficiency of operation.

TUNING FUNCTION - See TUNING.

TURNAROUND TIME - The time required between the entry or submission of a job in a computer system and the receipt of the completed results.

UNCONDITIONAL BRANCH - A branch action which always occurs regardless of conditions. Also see CONDITIONAL BRANCH.

UNPACK - To restore the original form of data which had previously been packed. Also see PACK.

UPDATE - The action of revising data in a master file or creating a new master file to reflect current conditions, status or corrections.

UTILITY PROGRAM - A computer program which provides general support for a computer system, such as a sort or trace program.

UTILIZATION - The ratio of time spent by a computer system, component, or device performing work during a specified interval, to the total time available, at a given workload.

VALIDATION - The evaluation, integration, and test activities performed at the system level to insure that a computer system being developed satisfies the requirements of a System Specification.

VARIABLE-LENGTH RECORD - A record that does not have a specified length.

VERIFICATION - The iterative process of determining whether the product of each step of the Computer Program Configuration Item (CPCI) development process fulfills all of the requirements specified by the previous step.

VIRTUAL MEMORY - See VIRTUAL STORAGE.

VIRTUAL STORAGE - A technique which permits a user to regard secondary memory as an extension of main memory and thus provides the user with an apparent larger main memory than actually exists. Same as VIRTUAL MEMORY.

WEAPON FIRE CONTROL APPLICATION - The automated processes involved in directing the positioning and firing sequence of one or more weapons based on target tracking data analysis.

WORD - A set of characters considered as a unit.

WORD LENGTH - The number of bits in a word.

WORK - The sequence of operations required to be performed by a computer system to obtain the specified results of a designated amount of data manipulation.

WORK BREAKDOWN STRUCTURE (WBS) - A product-oriented family tree composed of hardware, software, services and other work tasks that defines the products to be developed or produced and relates the elements of work to be accomplished to each other and to the end product.

WORKING STORAGE - A section of main storage designed for the temporary storage of data while it is being used by an operation.

WORKLOAD - The amount of work performed by a computer system with a designated configuration, in a defined environment, and within a specified period of time. Also see WORK.

ZERO SUPPRESSION - The elimination of nonsignificant zeros from the left and right sides of a numeral.

INDEX
(Volume I)

acquisition, 2,4-6,8,9,15,16,18,21,27,29-32,45,49,51,56,57,59,61; Tab A-2
 (also see Vol. II)
 life-cycle, 3-6,10,14,16,20,21,27,30,31,38,59,61 (also see Vol. II)
 management, 1 (also see Vol. II)
 phases, 17,35
 process, 5,6,8,9 (also see Vol. II)
 strategy, 6,8
access, see Vol. II
 file, Tab A-24
 methods, see Vol. II
 random, see Vol. II
 sequential, see Vol. II
 speed, 28
 time, 26 (also see Vol. II)
AFR 800-14, 6,16,23,60 (also see Vol. II)
 800-2, see Vol. II
algorithm(s), 4,60; Tab A-14 (also see Vol. II)
analogy(ies), 4; Tab A-1,2,3,7,9,11,12,17 (also see Vol. II)
analysis, 15,24,44-46,55-57; Tab A-3,4,13,17,18 (also see Vol. II)
 automated, see Vol. II
 macro-, see Vol. II
 mathematical, 32
 performance, 42 (also see Vol. II)
 phase, 24
 sensitivity, 45; Tab A-21 (also see Vol. II)
 systems, see Vol. II
 techniques, 55 (also see Vol. II)
 tradeoff, 60
 worst-case, 43
architecture, 13 (also see Vol. II)

baseline(s), 5,6,14,31
 allocated, 8,17,18
 functional, 8
 structure, 9
benchmark(s) (ing), 1,3,59; Tab A-3,23,24 (also see Vol. II)
 program, 31
 synthetic, Tab A-21
 workload 31; Tab A-23
boundaries, 24,39,41,44,49; Tab A-17
buffer(s), 12; Tab A-15,18 (also see Vol. II)

capabilities
 competing, 29
 data processing, Tab A-25
 expansion, 13

- functional, 12
- future, see Vol. II
- override, 12
- capacity, 12,19,24,59-61; Tab A-3,7 (also see Vol. II)
- Central Processing Unit (CPU), Tab A-24 (also see Vol. II)
- code(s)(ing), 18,27,41,43,50,56
 - construction, 27
 - lines of, 43,45,56 (also see Vol. II)
 - object, 19,59,60 (also see Vol. II)
 - source, 45,60 (also see Vol. II)
- command and control, 2 (also see Vol. II)
- Command, Control and Communications (C3), 1,3,4,11,14,15,16,20,21,24,30-32,48,51,52,59 (also see Vol. II)
- competition, 5,8,17
- compiler(s), 13,60
- complexity (ies), 5,13,30,43; Tab A-3,16 (also see Vol. II)
- computer
 - configuration(s), 12; Tab A-18
 - engineering, see Vol. II
 - instructions, see Vol. II
 - performance evaluation, 59
 - program, 24 (also see Vol. II)
 - detail specification, 52
 - development plan, 52,62
 - flow charts, 53
 - sizing and timing data, 53,62
 - resources, 2,8,16 (also see Vol. II)
 - scientist(s), 3,4,35 (also see Vol. II)
 - selection, see Vol. II
 - system(s), 1-6,8,15,20,23-25,39,48,51; Tab A-1,2,7,9,11-13,18,21-23,25-27 (also see Vol. II)
 - time, 27,52
- conceptual phase, 6,8,16,17,27,60,62
- configuration(s), 12-14,21,24,28,41,49,52,53; Tab A-8,9,18 (also see Vol. II)
- Contract Data Requirements List (CDRL), 49,51
- cost, 2,3,15,17,23,28-30,39,41,42,60 (also see Vol. II)
 - estimating (ion)
 - relationships, see Vol. II
 - study, see Vol. II
 - techniques, see Vol. II
- Cost Performance Report (CPR), 29,51
- Cost/Schedule Control Systems Criteria (C/SCSC), 29,43,51
- cycle time(s), 26 (also see Vol. II)
- data base, 2,15,20,27,52,56 (also see Vol. II)
- Data Item Description (DID), 15,52,61
- Defense Systems Acquisition Review Council (DSARC), 5 (also see Vol. II)
- delay(s), 26,27,56; Tab A-4

- Department of Defense (DoD), 2,5,51 (also see Vol. II)
 - Directive, 2 (also see Vol. II)
 - Instruction, 5,51
 - Management Steering Committee, 2
- Design Review
 - Critical (CDR), 18,24
 - Preliminary (PDR), 18,24
 - System (SDR), 8,17
- disc(s), Tab A-24
- documentation, 20
- Electronic Systems Division (ESD), 3,4,15,17,21,35,42,50,52,61,62; Tab A-31
 - (also see Vol. II)
 - ESD C3 Major Projects, (see Vol. II, Tab A1 for the following)
 - AABNCP 481B
 - AFSATCOM I 1205
 - AWACS 411L
 - COBRA DANE 633A
 - COMBAT GRANDE
 - CONUS OTH-B 414L
 - JSS 968H
 - JTIDS/ASIT 634B
 - NCMC 427M
 - PAVE PAWS 2059
 - SACDIN 1136
 - TACSI 485L
 - TRI-TAC
 - WWMCCS
- embedded, see Vol. II
 - computer(s), 3
 - resources, 2,9 (also see Vol. II)
 - System(s) (ECS), 1,3-6,8,14,15,20,23,24,26,32,39,48,51 (also see Vol. II)
 - sizing and timing estimates, 3
- empirical
 - equations, Tab A-1,13
 - methods, see Vol. II
 - studies, see Vol. II
- environment, 1,11,19,45,59 (also see Vol. II)
- equations, see Vol. II Tab B
- equipment, 45; Tab A-1,7,9 (also see Vol. II)
- estimates, 2,14-17,20,24,27,28,31,32,37,42,43,45,47,48,50,53; Tab A-7,8,
 - 9,15 (also see Vol. II)
 - sizing and timing, 1,3,6,9-11,15,17,19-21,29-31,38,45,52,55,56,59,60,62
 - (also see Vol. II)
- estimating, 2,56,61 (also see Vol. II)
 - methodologies, 2
 - resource, 2,27,39 (also see Vol. II)
 - sizing and timing parameters, 3-5,12-14,19,35,36,42,53,59; Tab A-1,2,7,
 - 9,11-13,17,24 (also see Vol. II)
 - techniques, 17,44 (also see Vol. II)

- FEDSIM, 21
- file
 - access, Tab A-24
 - organization, see Vol. II
 - size, 17 (also see Vol. II)
 - structure, 13,17
- firmware, 12,39,56,59
- Full-Scale Development, 8,14,17,27,32,61,62
- function(s), 1,4,12,15,23,30,41,47; Tab A-1,3,4,7,8,13,17 (also see Vol. II)
- hardware, 3,9,14,16,21,24,28,31,39,59; Tab A-3,9,12,25 (also see Vol. II)
 - characteristics, 40 (also see Vol. II)
 - configurations, 28
 - measurements, see Vol. II
 - monitors, 12; Tab A-4,25,26 (also see Vol. II)
 - parameters, 39
 - requirements, 12,13 (also see Vol. II)
 - size, 27
 - uncertainties, 1
- hardware and software
 - acquisition process, 6
 - characteristics, 2
 - development, 3,9
 - monitors, Tab A-4,27
 - needs, 1
 - problems, 3
 - requirements, 2
 - technologies, 1
 - tradeoffs, see Vol. II
 - utilization, 12
- Input/Output (I/O)
 - device(s), 28,30
 - handlers, 13
 - rates, 12
 - speeds, 13
 - terminals, 19 (also see Vol. II)
- instructions, see Vol. II
- interpret, Tab A-25-27
- language(s), see Vol. II
 - COBOL, see Vol. II
 - FORTRAN, see Vol. II
 - High Level (HLL), 60
 - High Order (HOL), see Vol. II
 - JOVIAL, see Vol. II
 - Machine-Oriented (MOL), see Vol. II
 - programming, 13
 - sizing, see Vol. II
- Life-Cycle Cost (ing) (LCC), see Vol. II

- management, 2,5,46 (also see Vol. II)
 - approach, see Vol. II
 - procedures, 2,3
 - reporting systems, 29
 - system(s), 43,51
 - techniques, 2
 - technology, 5
- manloading, see Vol. II
- manpower, 27 (also see Vol. II)
- measurement
 - techniques, see Vol. II
 - tools, see Vol. II
- memory, 20,60,61; Tab A-2,4,26 (also see Vol. II)
 - access
 - speed, 28
 - time, 26
 - capacity, 59-61; Tab A-3,7 (also see Vol. II)
 - configuration, 13
 - core, 60
 - estimates, 24
 - expansion capability, 13
 - main, 19,24,60,61
 - read-only (ROM)
 - reserve, 60,61
 - size, see Vol. II
 - spare, 59,60
 - virtual, see Vol. II
- metrics, 2 (also see Vol. II)
- MIL-STD
 - 483, 17
 - 881A, 51
 - 1521A, 17
 - 1679, 60
- model(s)(ing), 1,3,27,42; Tab A-1,3,15,19-22 (also see Vol. II)
 - analytic (al), 32 (also see Vol. II)
 - deterministic, Tab A-19,20
 - piece, Tab A-15,16
 - probabilistic, Tab A-1,20
 - simulation, 32 (also see Vol. II)
- modern programming practices, 3
- module(s), 13,19,20,23,30,43,56 (also see Vol. II)
- monitor(s)(ing), 1,3,14,18,29,32,42,56,57,59,61,62; Tab A-4,25-27 (also see Vol. II)
 - developmental, 4,38,51-53
 - hardware, 12; Tab A-4,25-27
 - hybrid, Tab A-4,27
 - software, 13; Tab A-4,25-27
- multiprogramming, 12

Office of Management and Budget (OMB), 5
 Circular A-109, 5,9
 overlay, see Vol. II

parallel processors, see Vol. II

parameters, 15,20,26,28,32,39,40,45,47,48,57; Tab A-12,18-22,24 (also see Vol. II)
 hardware, 39
 sizing, 19,29,32,38,41,48,53,57,60,62; Tab A-9,12,17,26,27
 timing, 19,29,32,38,42,48,53,57,62; Tab A-9,12,17,23,26,27

performance, 1,21,30,45,56; Tab A-4,15,17,23,26,27 (also see Vol. II)
 analysis, 42 (also see Vol. II)
 characteristics, 40; Tab A-25
 constraints, 30
 evaluation, 31,59 (also see Vol. II)
 improvement, 31 (also see Vol. II)
 indicators, 28
 measures, 26,59 (also see Vol. II)
 monitoring, 42
 parameters, 28
 requirements, 16,23; Tab A-17 (also see Vol. II)
 system, see Vol. II

prediction set, 53-55,57

procedures, 4,9,18,20,21,36,38,48,51,61,62; Tab A-14 (also see Vol. II)
 application of, see Vol. II
 engineering and management, see Vol. II
 examples of, see Vol. II Tab C
 management, 2,3
 standard, 3,35
 study, 35,38,47

productivity, see Vol. II

programmer capability, see Vol. II

programming practices, see Vol. II

quality, 4,6,20,40,53; Tab A-8,10-12 (also see Vol. II)

queuing
 theory, Tab A-1,18 (also see Vol. II)
 time(s), 20

real-time, 1,12,59; Tab A-25 (also see Vol. II)

regulations, see Vol. II

relationship(s) (see Vol. II, Tab B for the following)
 applications vs. cycle time
 development time vs. core, months
 development time vs. core, man
 instructions vs. air targets tracked
 instructions vs. storage capacity/air target speed
 memory size vs. functions
 memory size vs. cycle time and functions
 message response time vs. input/output (I/O) terminals

- object words vs. functions
- storage capacity vs. air targets
- word size vs. cycle time
- reliability, 1,12,13,30,50; Tab A-1 (also see Vol. II)
- requirement(s), 12-17,21,23,28,30,35,42,44,49,53,61; Tab A-3,7 (also see Vol. II)
 - analysis, see Vol. II
 - core, 45,53
 - definitions, 12,13
 - ECS, 4,11,14
 - hardware, 12,13
 - memory, 60; Tab A-2
 - operational, 1,2 (also see Vol. II)
 - resource, 27
 - software, 13
 - storage, 23; Tab A-24
 - system, 11,21; Tab A-11,17
 - timing, 24,31,32
 - workload, 23
- response time, 20-29,31,47,48; Tab A-2,15 (also see Vol. II)
- risk(s), 4,8,9,43,55,57 (also see Vol. II)
- schedule(s)(ed)(ing), 15,30,38 (also see Vol. II)
 - method(s), 11
 - problems, 2
 - program, 27,53
- simulation(s)(ing), see Vol. II
- software, 1,2,8,9,13,14,17,19,23,24,27,28,31,40,59,60; Tab A-3,12 (also see Vol. II)
 - characteristics, 39
 - code counts, 43
 - construction, 8
 - development, see Vol. II
 - engineering, 5 (also see Vol. II)
 - growth, 43,56
 - integration, 3,18 (also see Vol. II)
 - management, see Vol. II
 - monitors, 13; Tab A-4,25-27
 - needs, 1
 - program, 42; Tab A-15
 - requirements, 13 (also see Vol. II)
 - size, 14,16,40,56,60
 - systems, 1
 - technologies, 1
 - testing, 18
 - uncertainties, 1
- Source Selection Evaluation Board (SSEB), 29

specification(s), 8; Tab A-3,23 (also see Vol. II)
 computer (see Vol. II, Tab A2 for the following)
 CDC Cyber 74, A2-3
 CDC Cyber 174-12, A2-5
 CDC System 17, A2-7
 CDC An/UYK-25 MP60, A2-9
 Data General NOVA 840, A2-11
 Data General NOVA 1220, A2-13
 DEC PDP 11/05, A2-15
 DEC PDP 11/10 and 11/40, A2-17
 Honeywell H-716, A2-19
 Honeywell H-6050 & 6060, A2-21
 Honeywell H-6080, A2-23
 IBM 370/155, A2-25
 Intel 80, A2-27
 Raytheon RDS-500, A2-29
 ROLM 1603, A2-31
 Texas Instruments TI-980A, A2-33
 UNIVAC AN/UYK-7, A2-35
 UNIVAC AN/UHK-20 (U-1600), A2-37
 UNIVAC U-1108, A2-39
 UNIVAC U-1110, A2-41
 UNIVAC U-1616, A2-43
 system, 8,10,16,17
 system segment, 8,10
 Statements of Work (SOW), 28
 storage, 24,48 (also see Vol. II)
 capacity, 60
 main, 39
 off-line, 13
 on-line, 13
 requirements, 23; Tab A-24
 study report, 43,46 (also see Vol. II)
 system(s)
 analysis, see Vol. II
 capacity, 60
 components, 40
 design, 2,11-13,23
 developments, 31 (also see Vol. II)
 discipline, 5
 implementation, 2
 operation(s)(al), 31,59; Tab A-4
 performance, 1,16,21,26,31,45; Tab A-17,26 (also see Vol. II)
 upgrade, 9

 techniques, 2-5,9,20,27,32,41,42,44,46,49,50 (also see Vol. II)
 analogy, Tab A
 analysis, 55
 analytical, Tab A
 benchmarks, Tab A

estimating, 17,44
models, Tab A
monitors, Tab A
simulations, Tab A
sizing and timing, 32; Tab A
terminals, 19,20,24,40; Tab A-15
testing, 18,40 (also see Vol. II)
throughput, 12,20,26-29,31; Tab A-2 (also see Vol. II)
time-sharing, 59
tradeoffs, 8,49,60 (also see Vol. II)
transfer rates, see Vol. II
trends, 15,40,43 (also see Vol. II)
turnaround time, 12

uniprocessors, see Vol. II
utilization, 12,20,26,39,60,61; Tab A-4 (also see Vol. II)

Verification and Validation (V&V), 4,24,59 (also see Vol. II)

weapon systems, 15,51 (also see Vol. II)
Work Breakdown Structure (WBS), 51 (also see Vol. II)
workload, 1,11,12,15,18,20,21,23,24,26-32,39-41,45,46,49,52,53,59,61,62;
Tab A-3,9,12,13,15,20,21-27 (also see Vol. II)

END

DATE
FILMED

8-80

DTIC